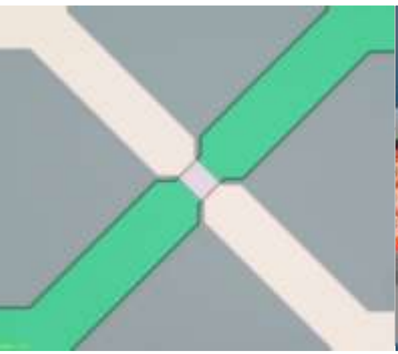
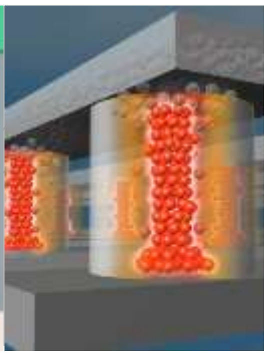
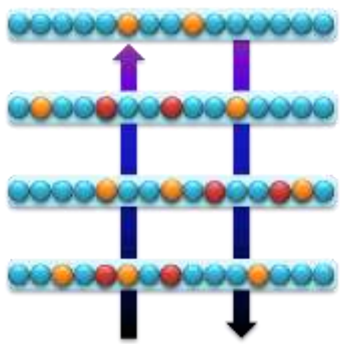
Exceptional service in the national interest

Sandia National Laboratories







HAANA
Hardware Acceleration of Adaptive Neural Algorithms

Designing an Analog Crossbar based Neuromorphic Accelerator

Sapan Agarwal, Alexander Hsia, Robin Jacobs-Gedrim, David R. Hughart, Steven J. Plimpton, Conrad D. James, Matthew J. Marinella
Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY | NNSA | LDRD

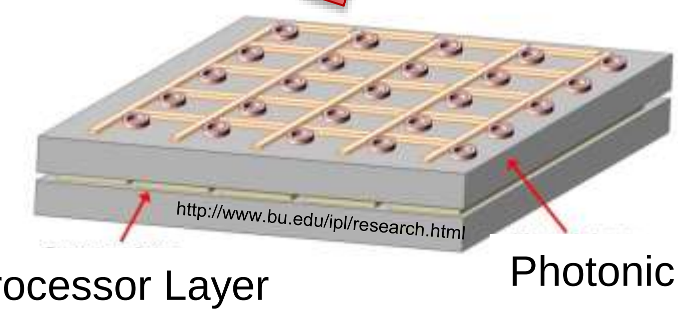
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

The Von Neumann Bottleneck

CPU — **Communications Bus** — **Memory**

Current Transistors ~ 10 aJ
40kT Noise Limit ~ 0.2 aJ

Cross chip communications ~ 1 pJ
DRAM Access >10 pJ
Ethernet ~ 1nJ



Communications require orders of magnitude more energy!

Optical interconnects 100 fJ to 1 pJ

Use Resistive Memories for Local Computation

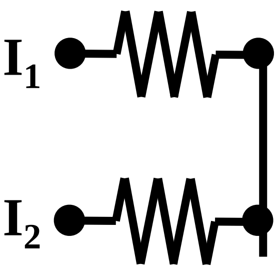
Sandia National Laboratories

$$V = I \times R$$

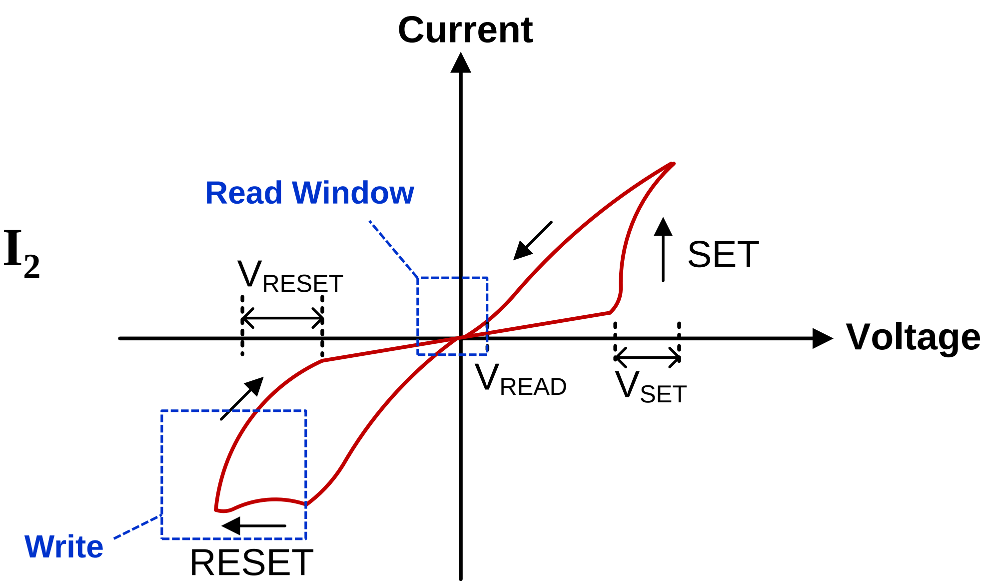
$$I = G \times V$$

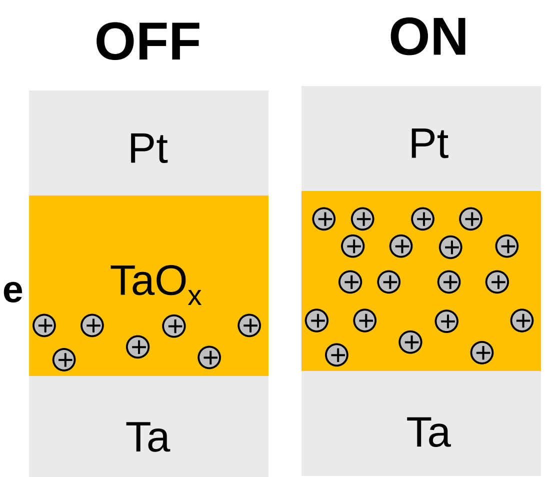
multiplication

- A resistive memory or ReRAM is a programmable resistor
  - Apply small voltages allows the conductance to be read: I = G × V
  - Apply large voltages to change the resistance

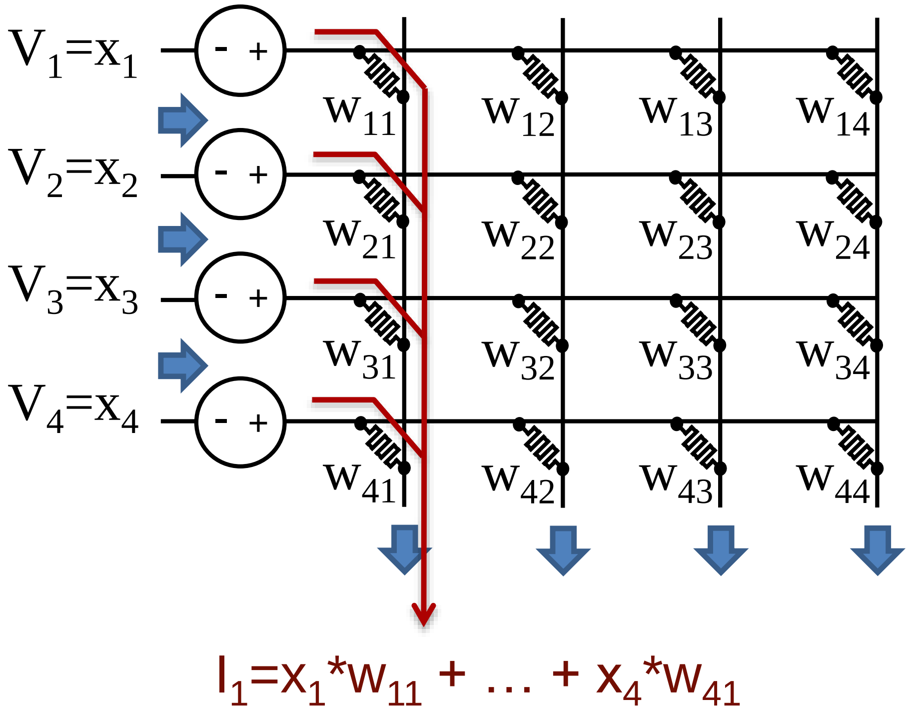


Addition: $I = I_1 + I_2$





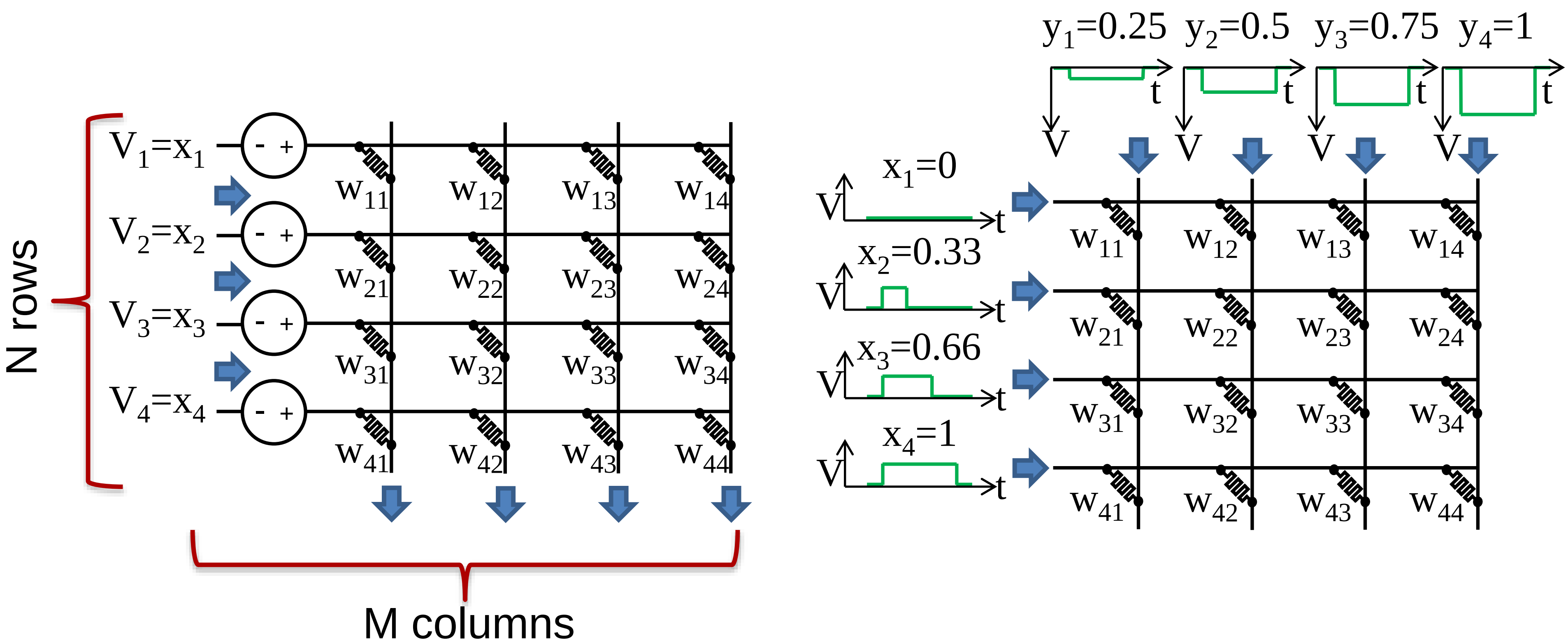
Directly Process in the Memory Itself



Analog is efficiently and naturally able to combine computation and data access

Effectively, large-scale processing in memory with a multiplier and adder at each real-valued memory location

Crossbars Can Perform Parallel Reads and Writes

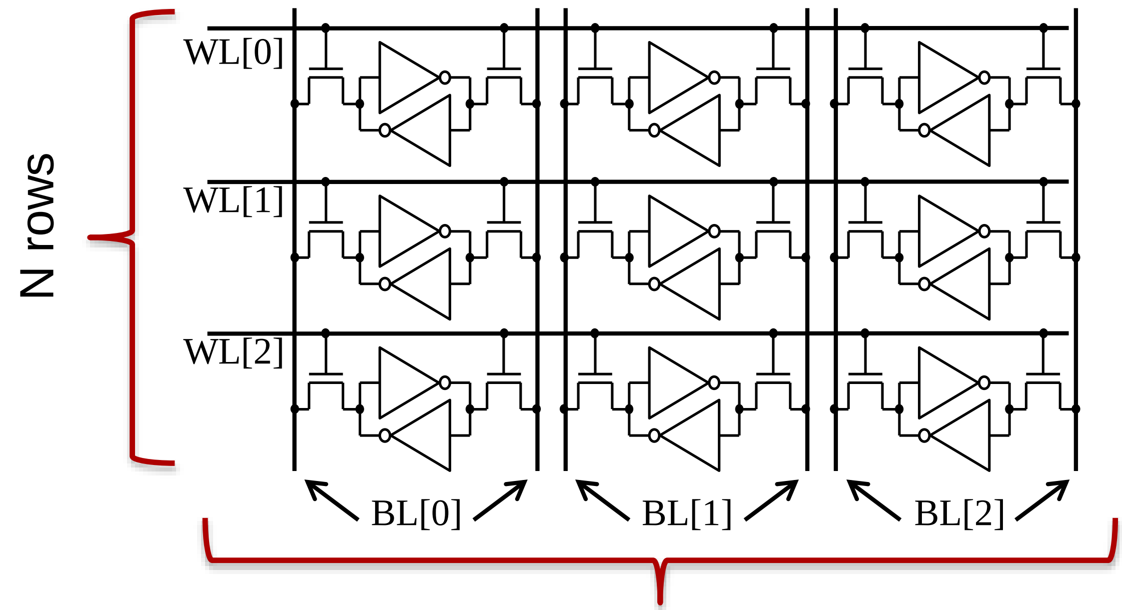


Energy to charge the crossbar is CV^2

$E \propto C \propto$ number of RRAMs $\propto N \times M$

$$E \sim O(N \times M)$$

SRAM Arrays Require Charging Columns Multiple Times



M columns

SRAMs must be read one row at a time, charging M columns

Each column wire length is O(N).

Energy = N Rows × M Columns × O(N) wire length

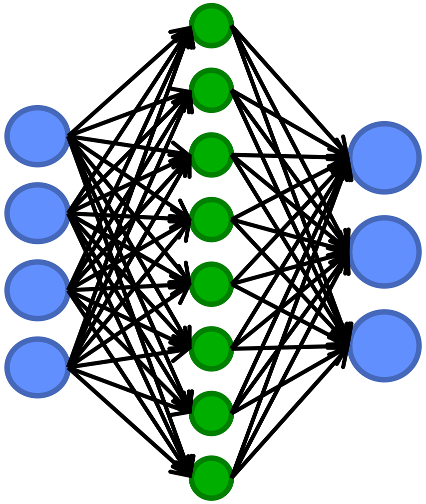
Energy ~ $O(N^2 \times M)$

O(N) times worse than a crossbar!

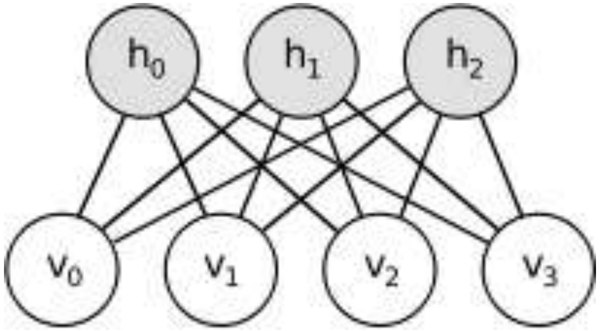
"For Internal E3S Use Only. These Slides May Contain Prepublication Data and/or Confidential Information."

Want To Accelerate Many Different Neural Algorithms

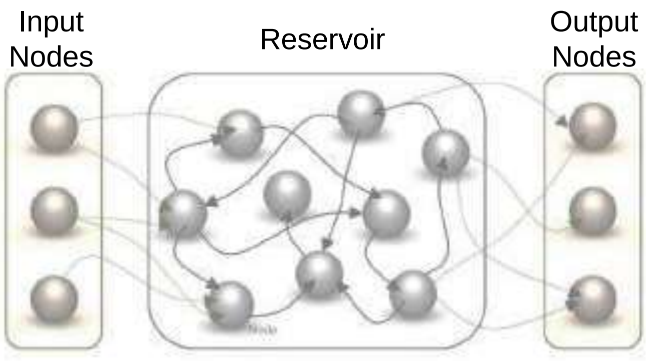
Backpropagation



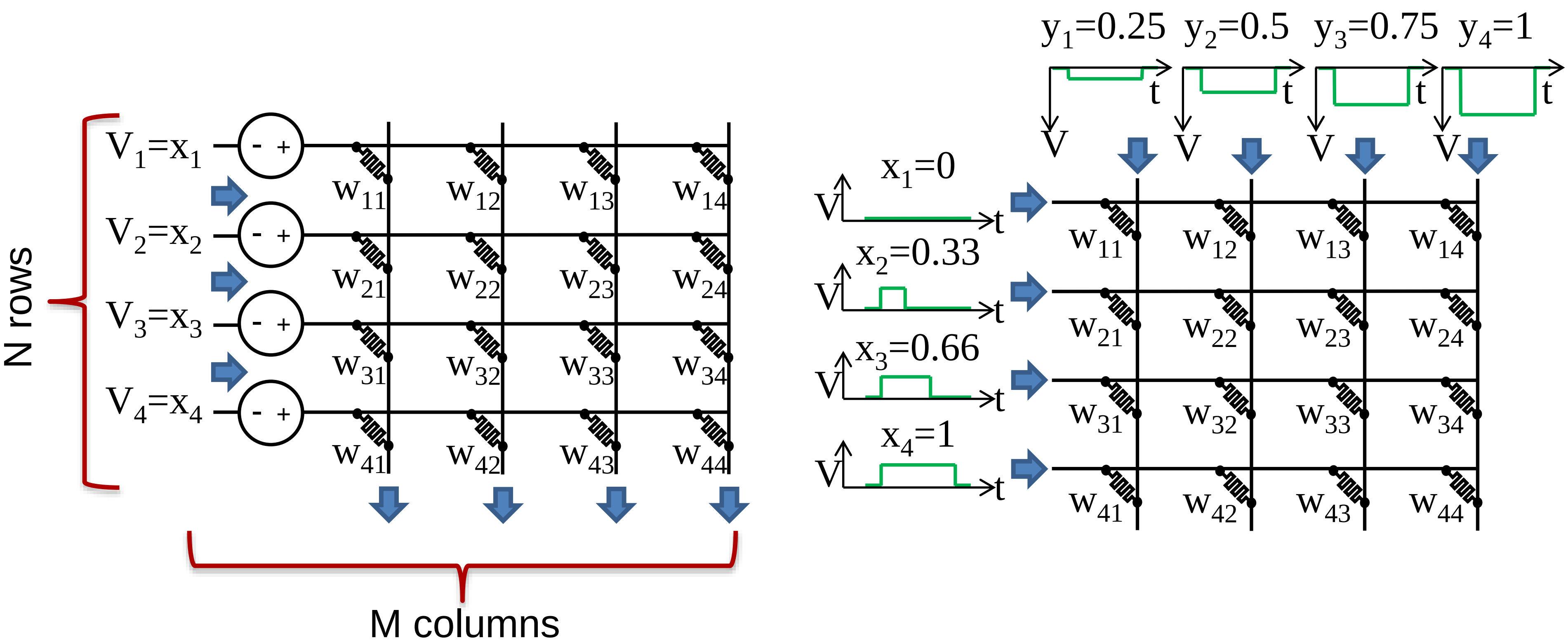
Sparse Coding



Liquid State Machine



Crossbars Can Perform Parallel Reads and Writes



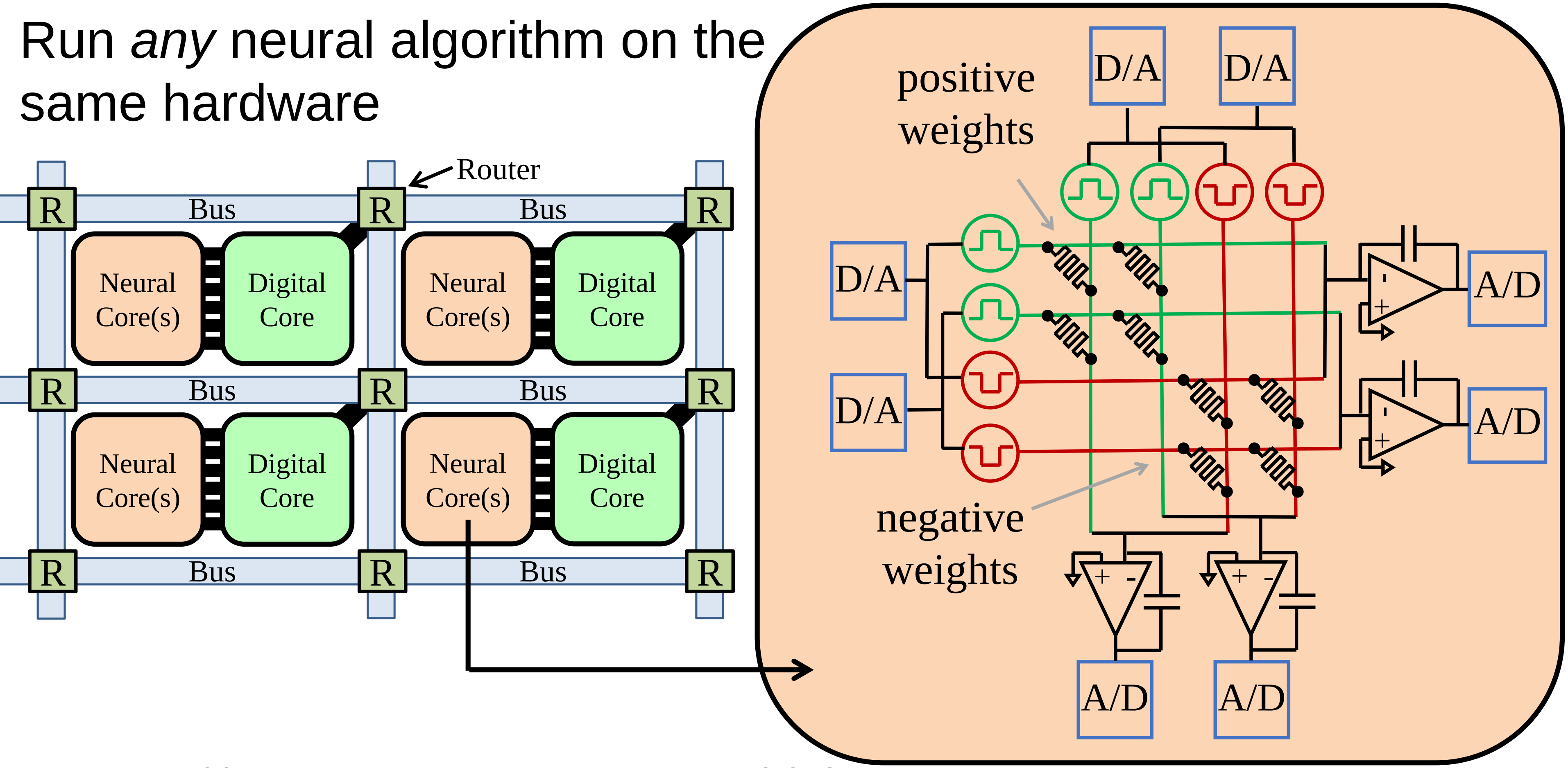
Energy to charge the crossbar is CV^2

$E \propto C \propto$ number of RRAMs $\propto N\times M$

$$E \sim O(N\times M)$$

General Purpose Neural Architecture

Run *any* neural algorithm on the same hardware



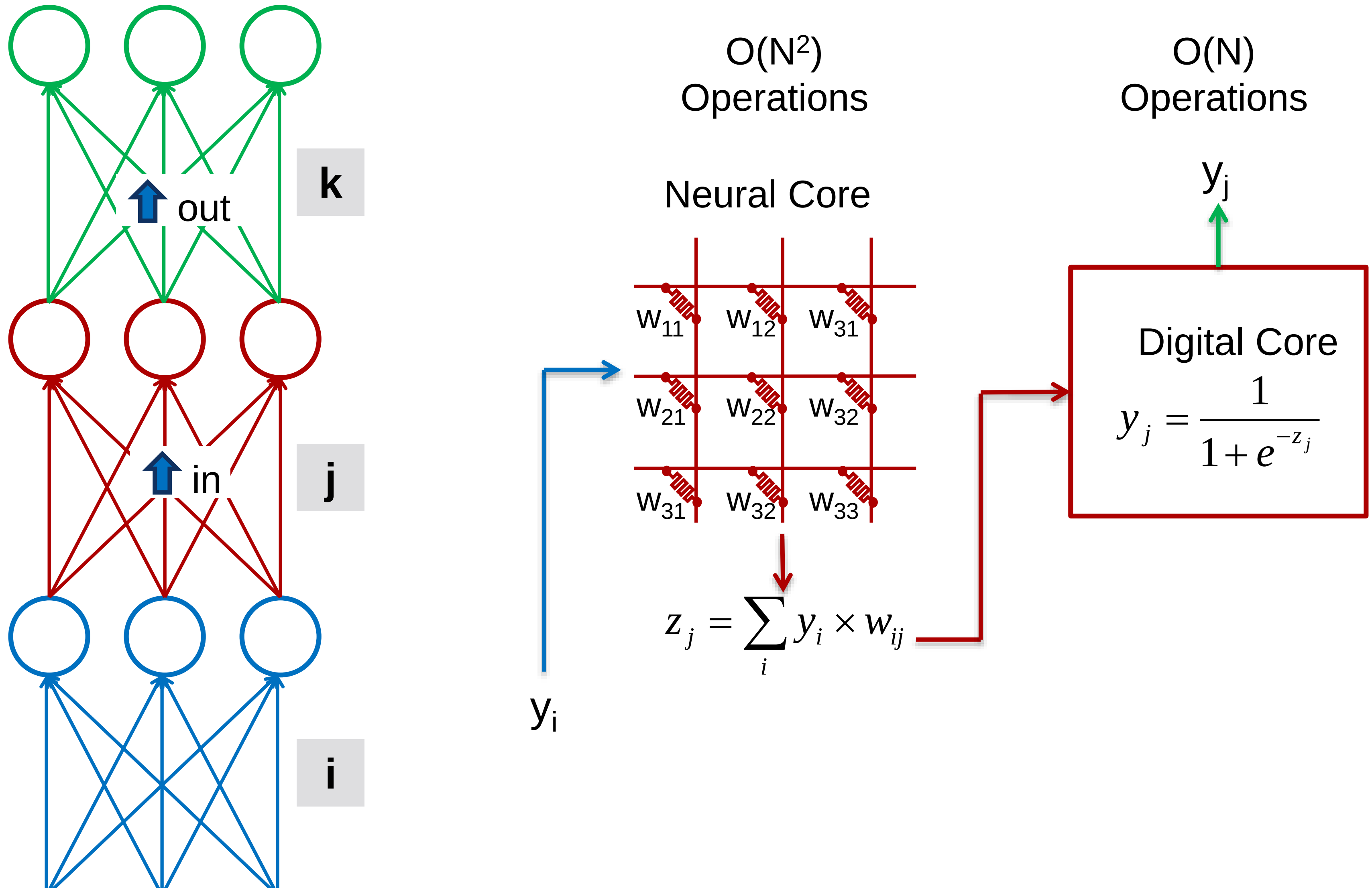
Neuromorphic core:

- Evaluate vector matrix multiplies along rows or columns
- Train based on input vectors

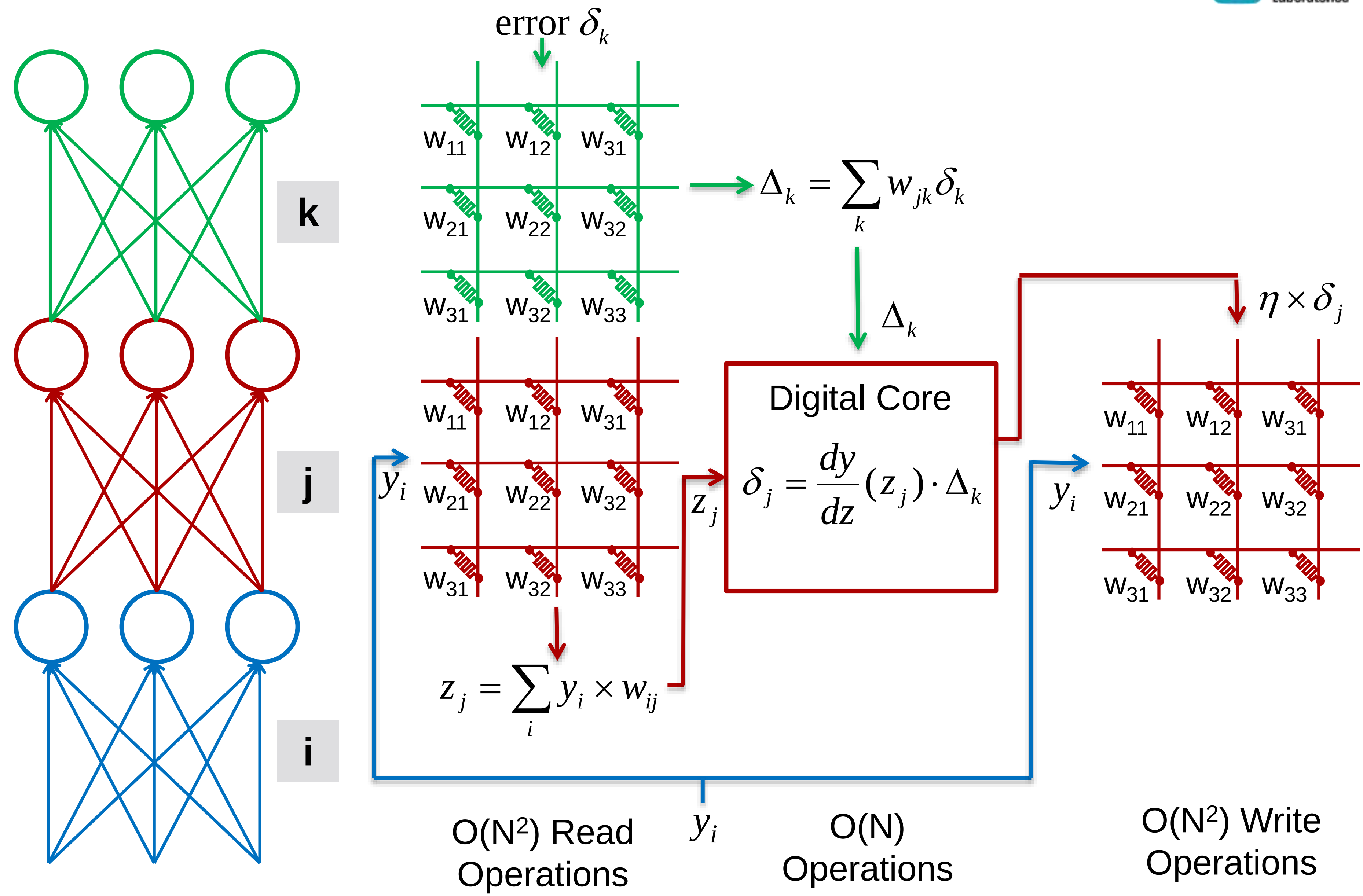
Digital Core:

- Process neural core inputs/outputs
- For NxN crossbar, the crossbar accelerates $O(N^2)$ operations leaving only O(N) operations for the digital core

Can Run Neural Networks on this Architecture



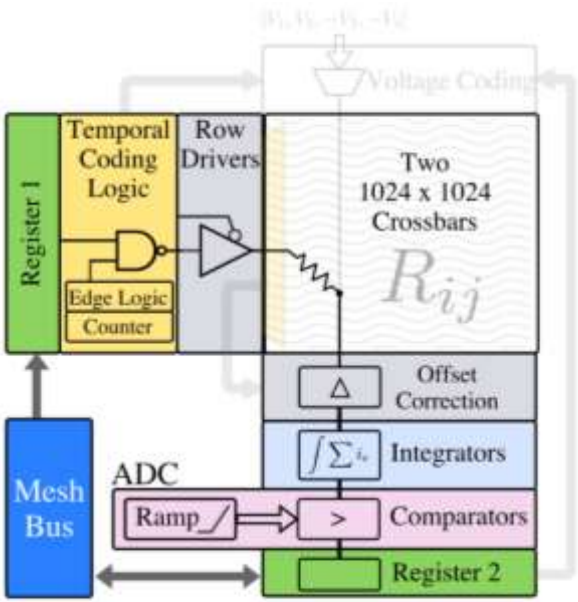
Back Propagation



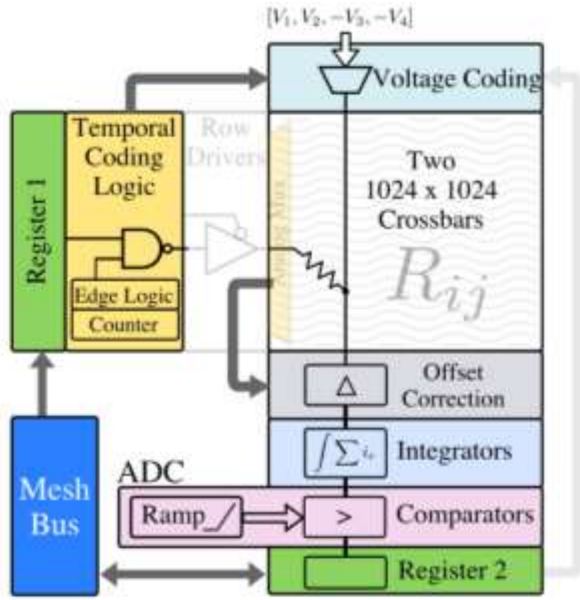
Design & Model Detailed Architecture

Sandia National Laboratories

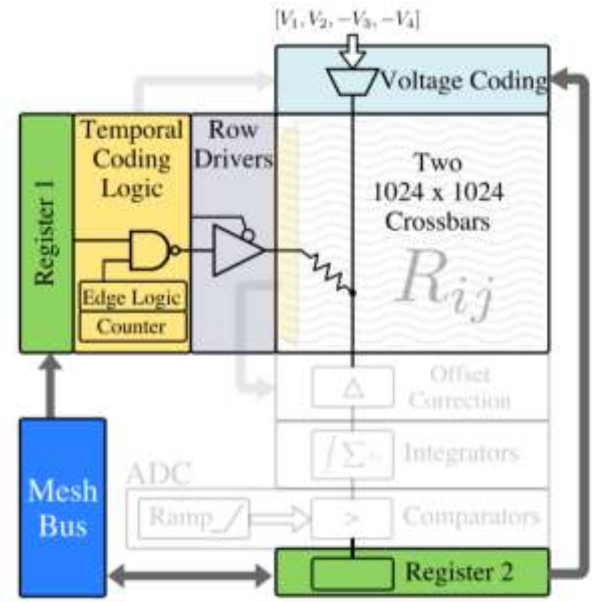
Vector Matrix Multiply



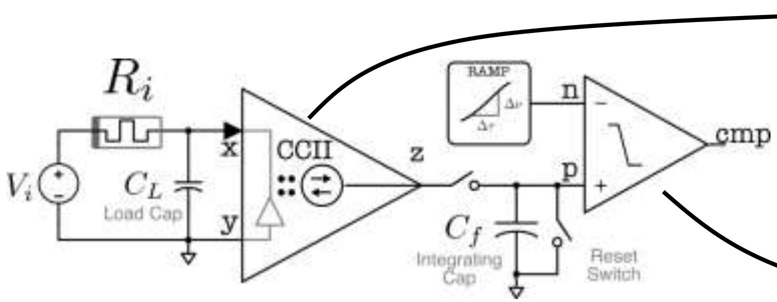
Matrix Vector Multiply



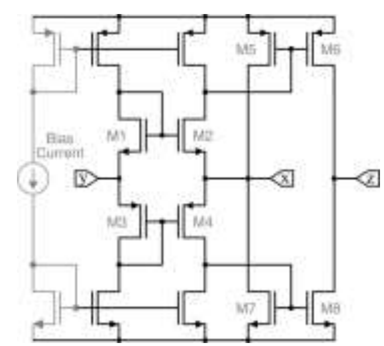
Outer product Update



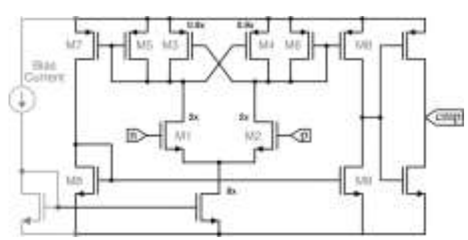
Neuron Circuitry



Current Conveyor Based Integrator

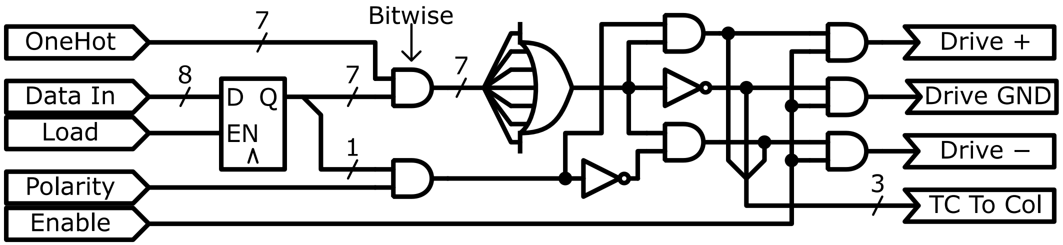


Comparator

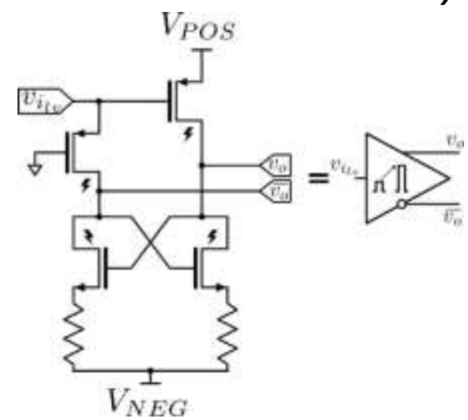


Row & Column Driver Circuitry

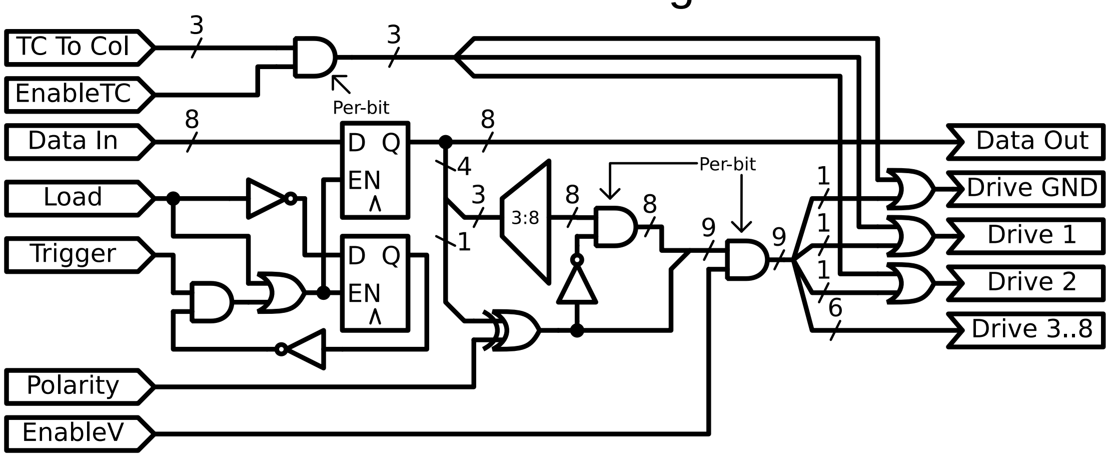
Row Driver Logic



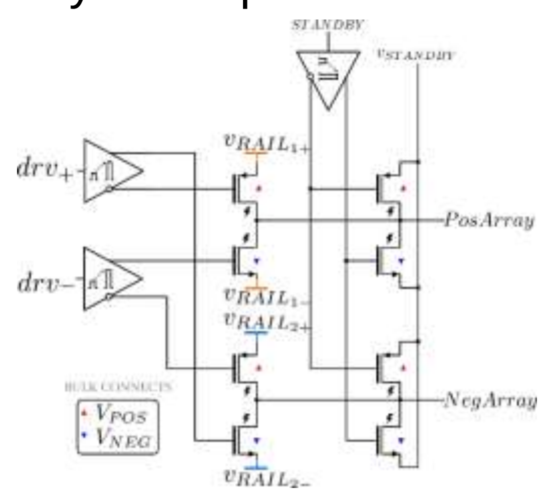
Voltage level shifter (drive high V transistor with low V)



Column Driver Logic



Array driver pass transistors



Compare Architectures

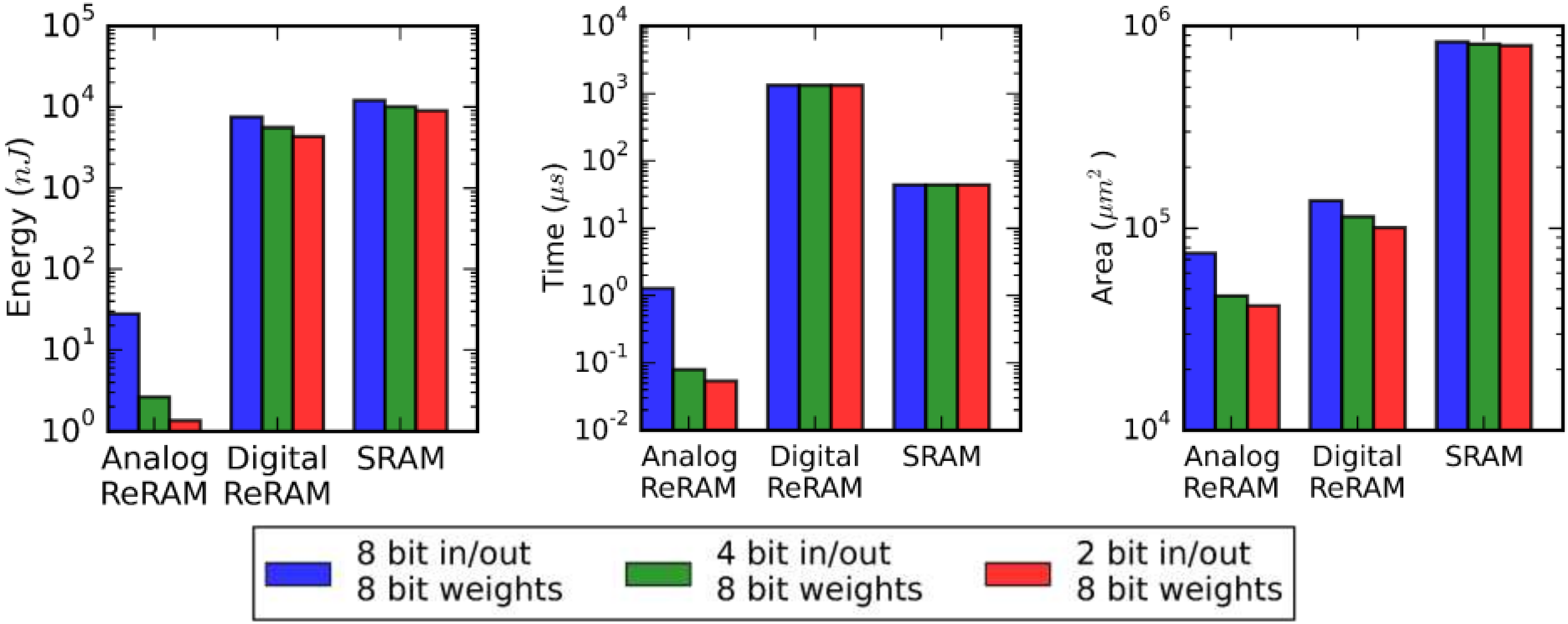
1024 x1024 = 1M array operations, sum over 1 training cycle, 3 operations:

- Vector Matrix Multiply
- Matrix Vector Multiply
- Outer Product Update

Energy
430 – 6,900X over SRAM

Latency
35 – 800X over SRAM

Area
11 – 20X over SRAM

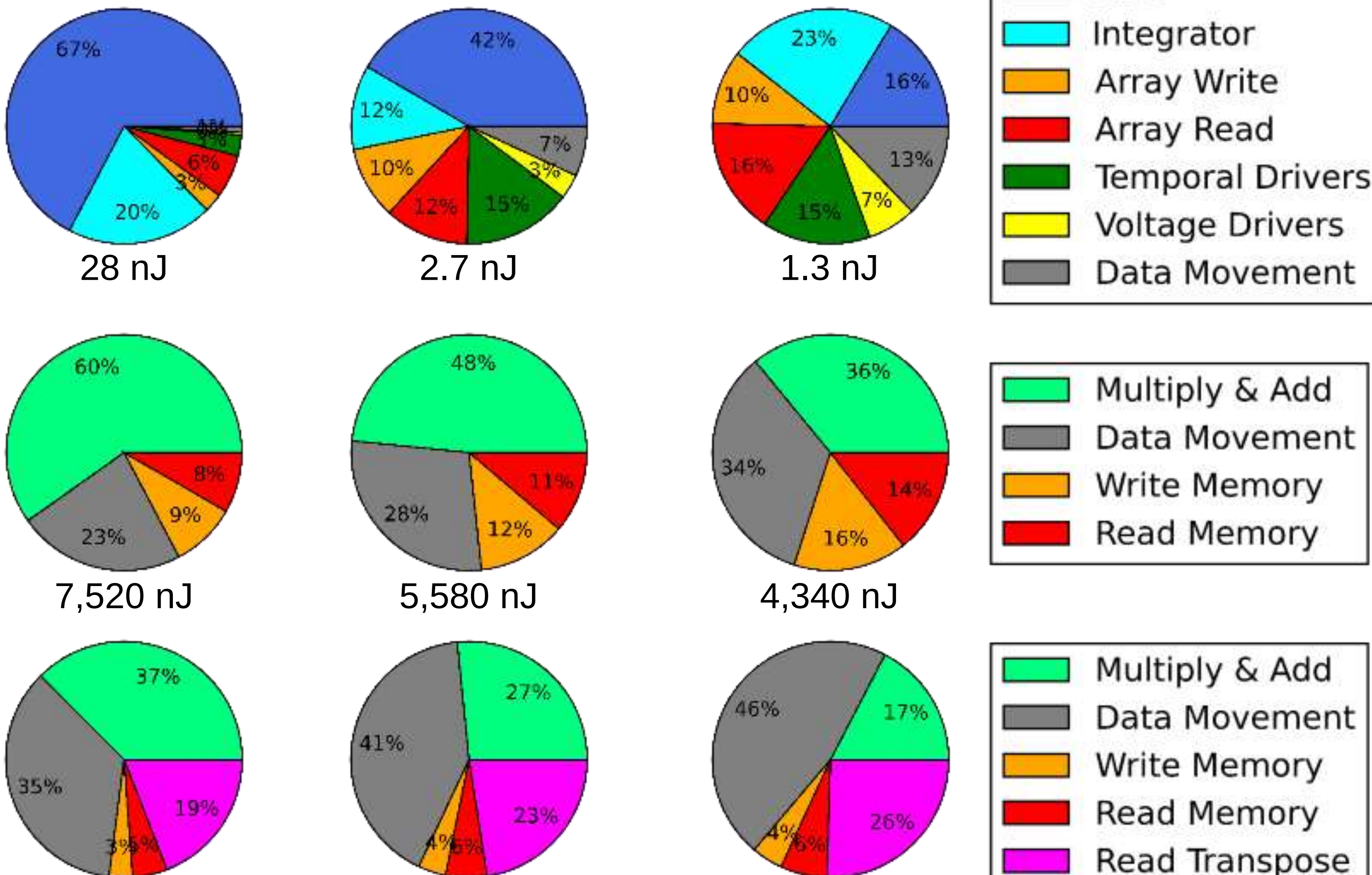


Used a commercial 14/16 nm PDK

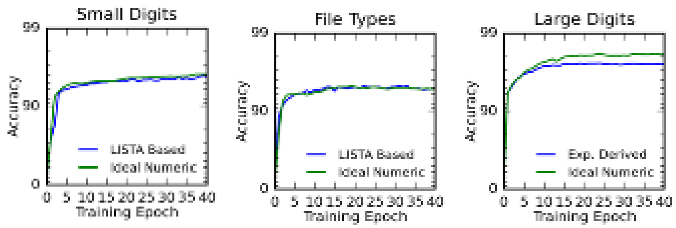
***Requires 100 MΩ on state devices

Neural Core Energy Analysis

	8 bits In/out 8 bit weights	4 bits In/out 8 bit weights	2 bits In/out 8 bit weights
Analog ReRAM	28 nJ	2.7 nJ	1.3 nJ
Digital ReRAM	7,520 nJ	5,580 nJ	4,340 nJ
SRAM	12,010 nJ	10,150 nJ	8,970 nJ



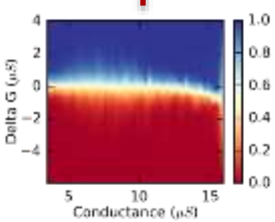
Multiscale Model of a Neural Training Accelerator



Target Algorithms

- Deep Learning
- Sparse Coding
- Liquid State Machines

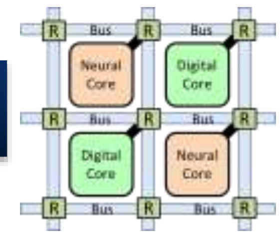
Algorithms



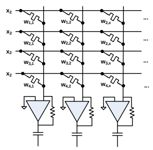
CROSS SIM

Sandia Cross-Sim: Translates device measurements and crossbar circuits to algorithm-level performance

Architecture



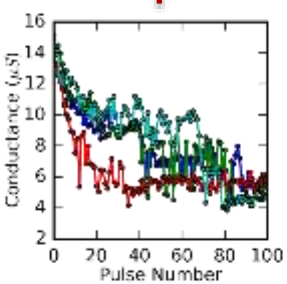
Modified McPAT/CACTI: Model performance and energy requirements



Circuits

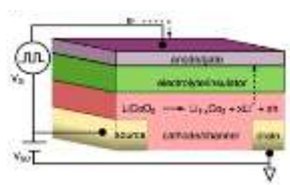
Xyce

Sandia's Xyce Circuit Sim: Simulate crossbar circuits based on our devices

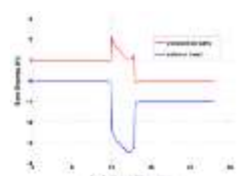


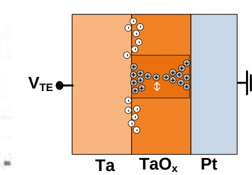
Memristor fabrication and measurements in MESAFab

Devices



Drift-diffusion model of ReRAM band diagram & transport (REOS, Charon)

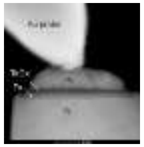


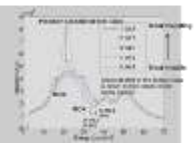


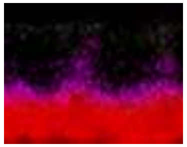
DFT of model of oxide physics, bands

Materials

In situ TEM of filament switching: Use DFT model to interpret EELS signature

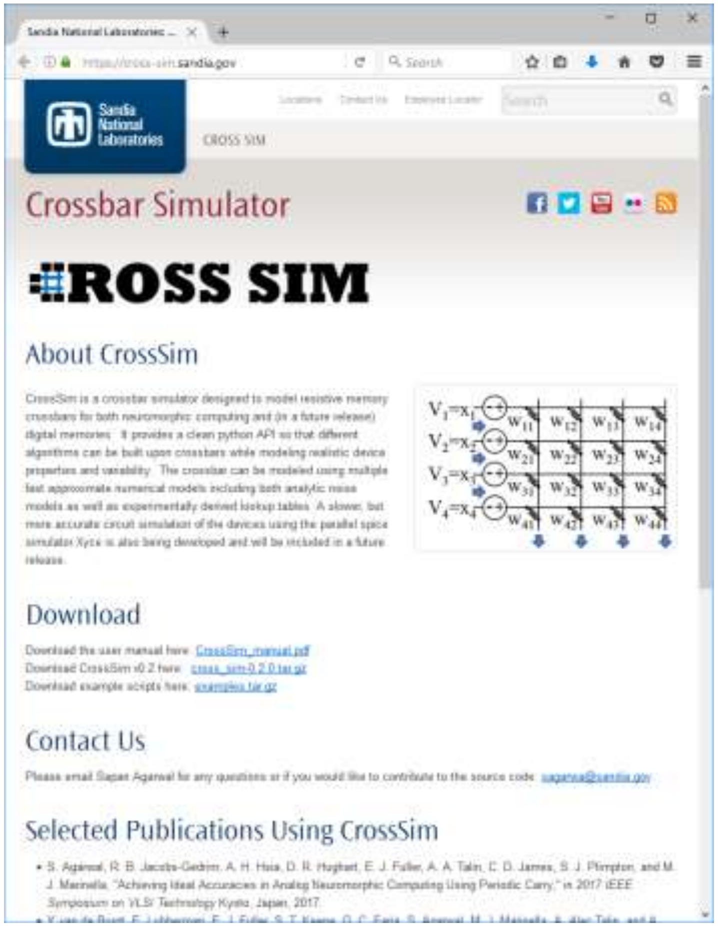


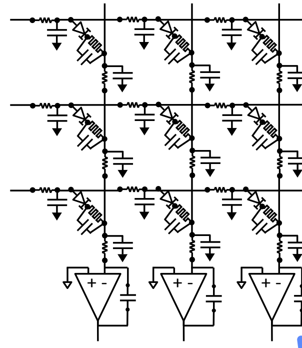




CROSS SIM

https://cross-sim.sandia.gov





Learning Algorithm

Neural Core Simulator

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix}$$

Xyce Crossbar Circuit Model

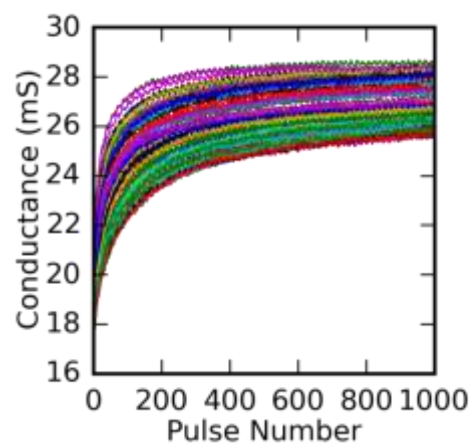
Physical Hardware Crossbar

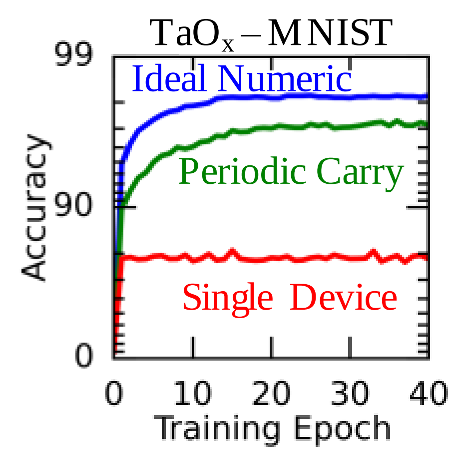
Numeric Crossbar Simulator

Detailed but slow

Fast but approximate

Measured Devices ⟷ **Algorithmic Performance**





Simple Python API:

```
# Do a matrix vector multiplication
result = neural_core.run_xbar_mvm(vector)
```

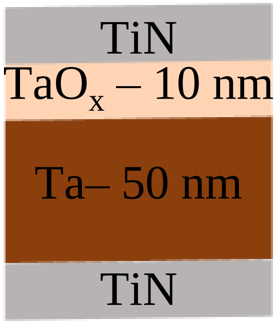
Simple API to model crossbars

Sandia National Laboratories

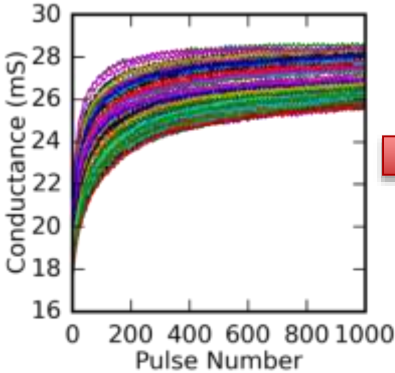
```
# ************** set parameters defining the crossbar

params.algorithm_params.weights.sim_type = "XYCE" # Use a XYCE based sim
params.algorithm_params.weights.maximum = 10  # clipping limits
params.algorithm_params.weights.minimum = -10  # clipping limits
params.xyce_parameters.xbar.device.TAHA_A1 = 4e-4  # Xyce Parameters
…
…

# ************** API for running neural operations
#                          All crossbar details are transparent to the user

# Create a neural_core object that models a crossbar
neural_core = MakeCore(params=params)

neural_core.set_matrix(weights) # set the initial weights
result = neural_core.run_xbar_vmm(vector) # Do a vector matrix multiply
result = neural_core.run_xbar_mvm(vector) # Do the transpose, a matrix vector mult.
neural_core.update_matrix(vector1,vector2) # Do an outer product update
```

Go from Measurement to Accuracy

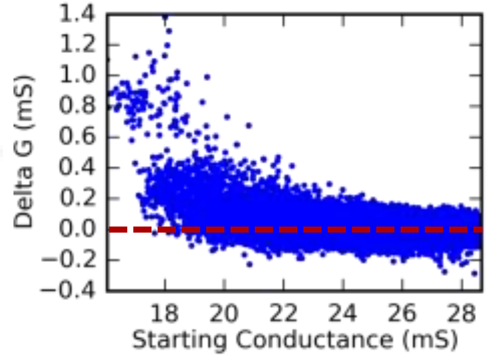
Fabricate Device



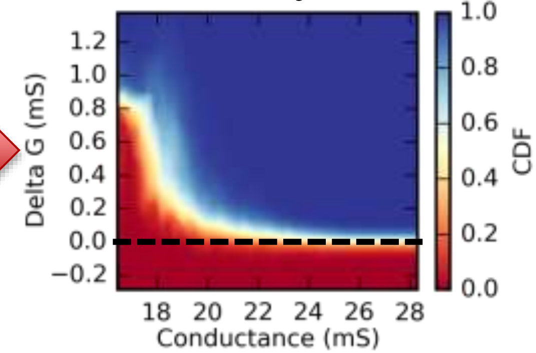
Measured Pulsing

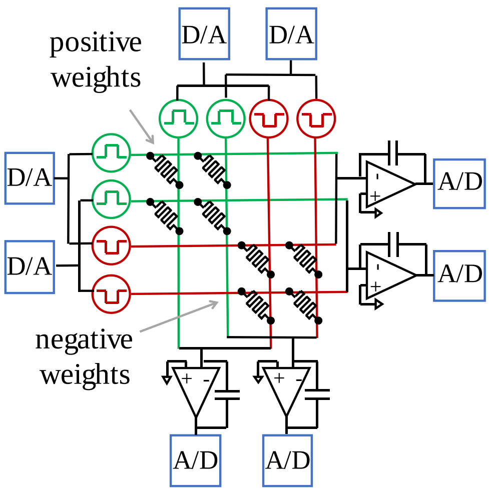


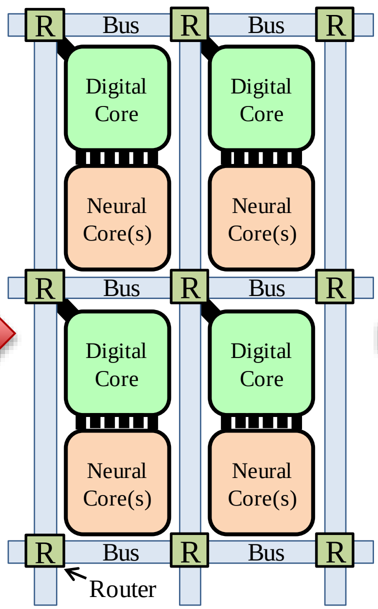
ΔG Scatterplot

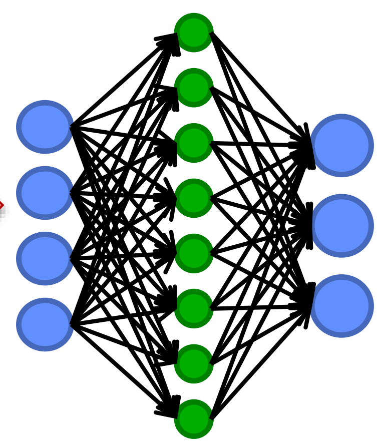


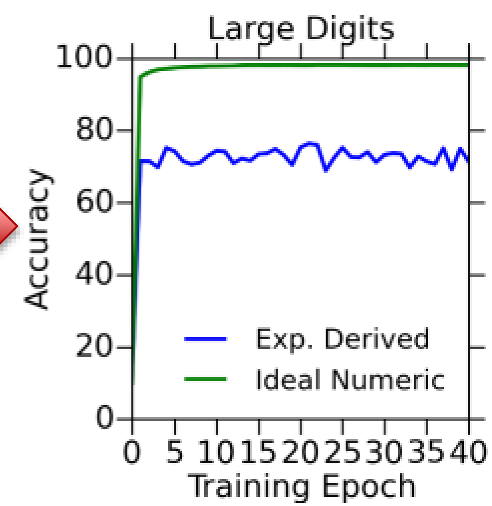
Cumulative Probability of ΔG







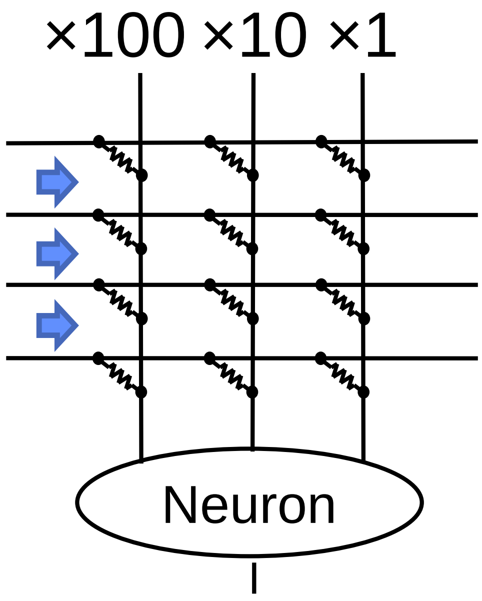




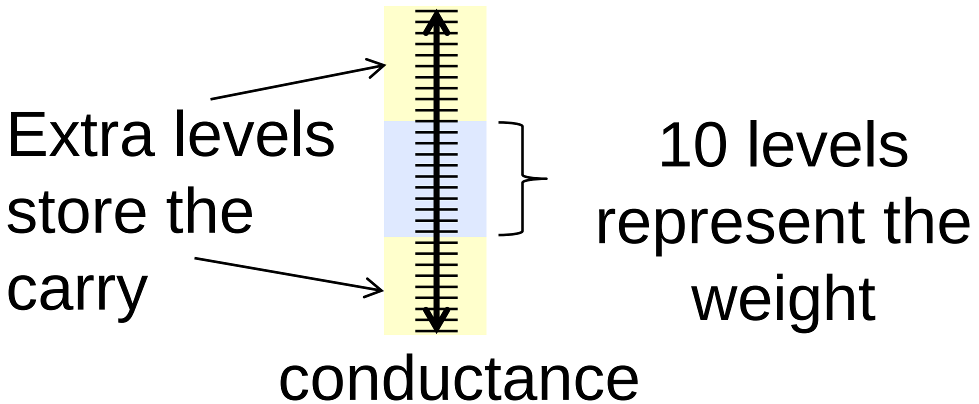
Multi-ReRAM Synapse: Periodic Carry

If we need more bits per synapse, use multiple memristors

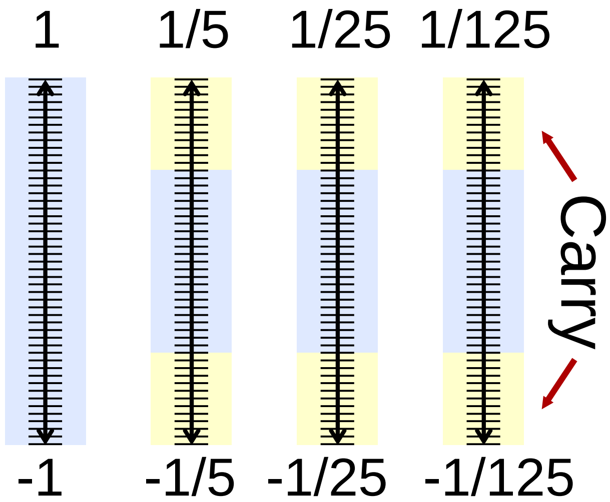
- Three 10 level ReRAMs could represent 1-1000!
- Adding to the weight requires reading every ReRAM to account for any carries and serially programming each ReRAM: VERY EXPENSIVE

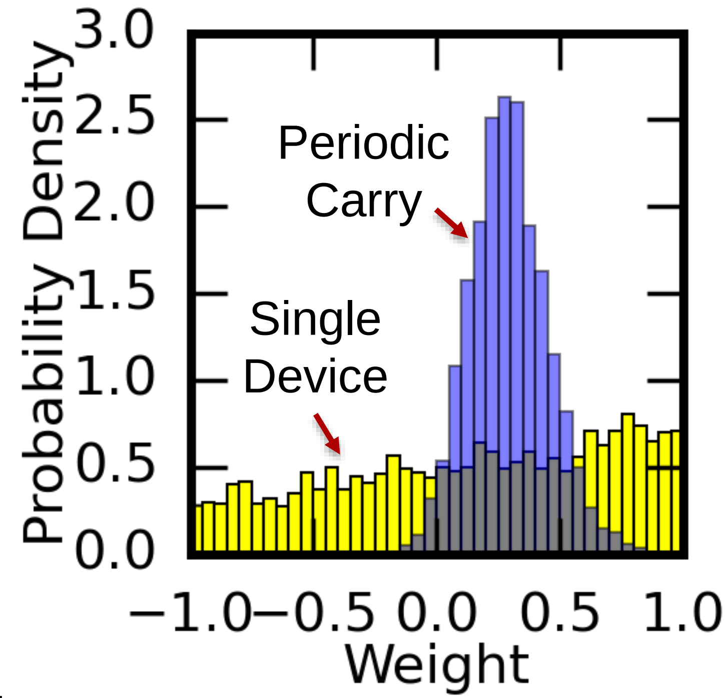


- Use >10 levels to represent a base 10 system
- Ignore carry and program the crossbar in parallel.
- Periodically (once every few hundred cycles) read the ReRAM and perform the carry



Periodic Carry Compensates for Write Noise





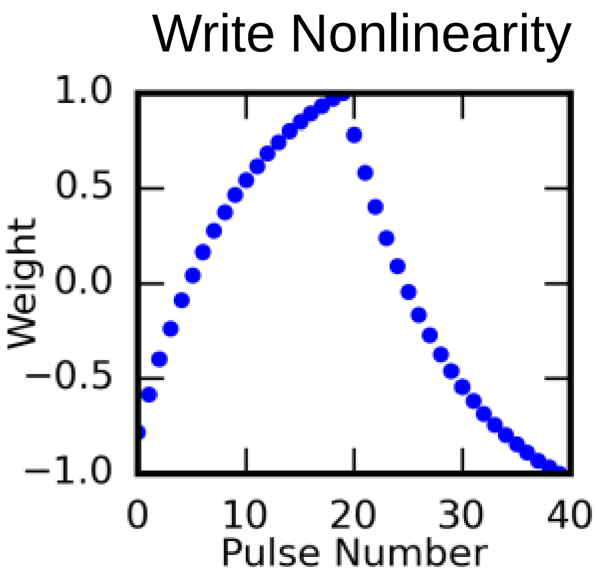
Read and reset every 100 pulses
Do 300,000 small (0.02% of weight range) updates
- net of 1500 positive training pulses

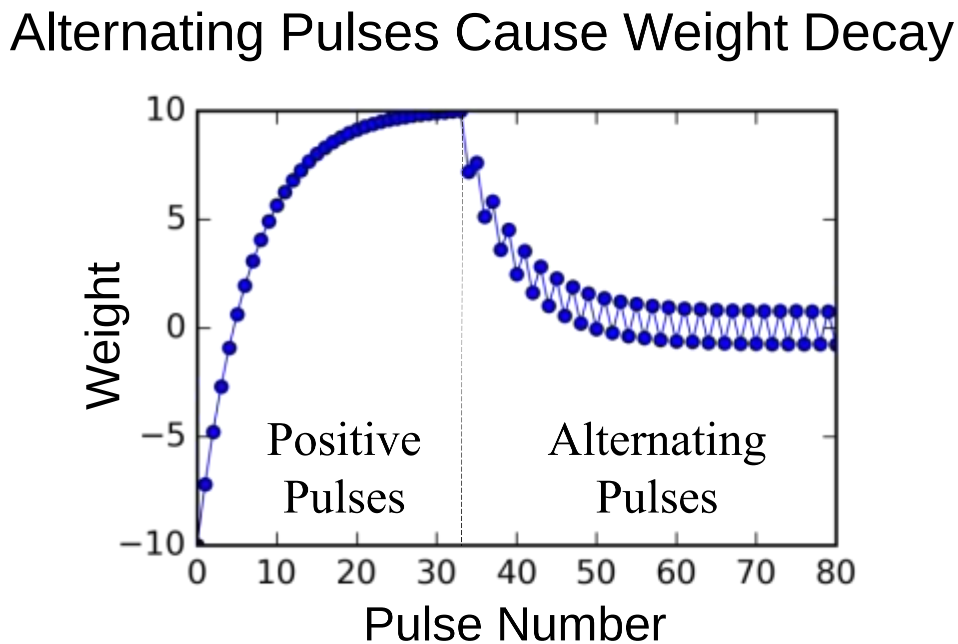
Learn from a 0.5% Signal

Noise Sigma = 1.4% for single device
- (from $\sigma_{noise} / G_{range} = 0.1\sqrt{\Delta G / G_{range}}$)
- Write noise applied during updates and carries

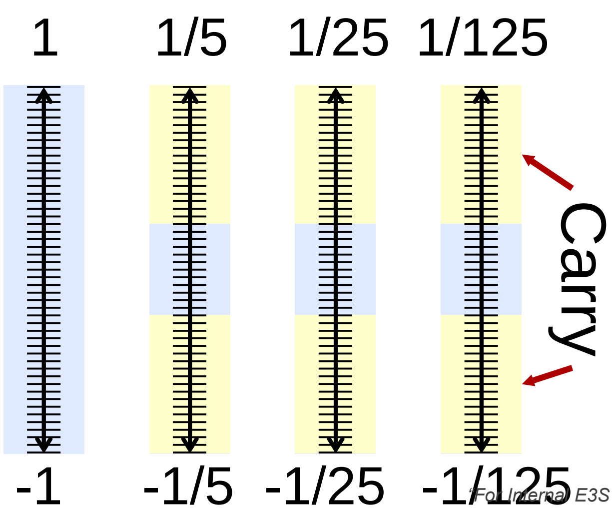
Periodic Carry Mitigates Write Nonlinearity

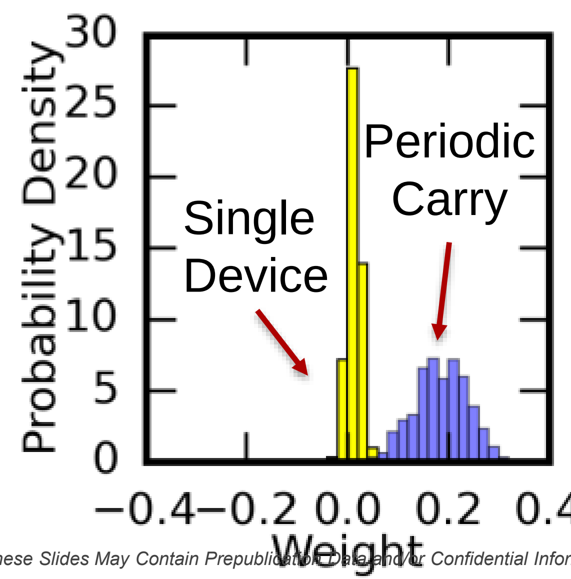
Sandia National Laboratories





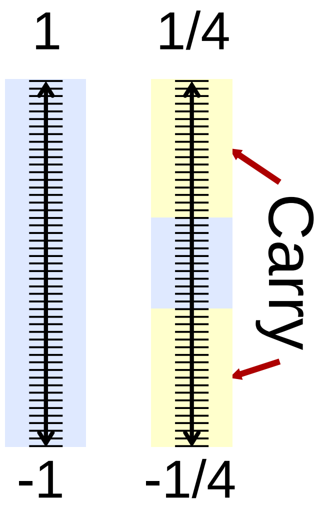
Use center linear range of weights



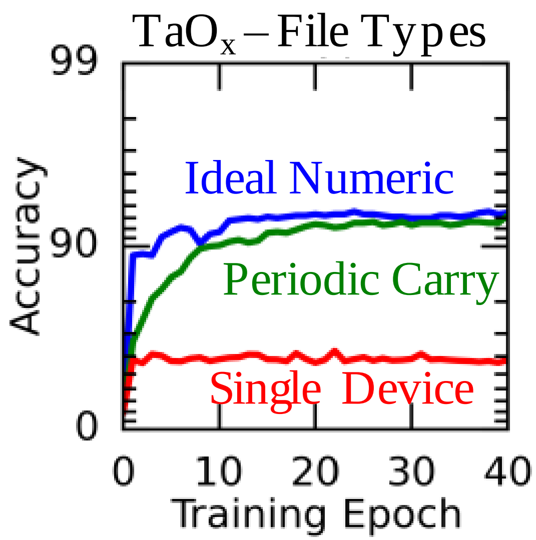


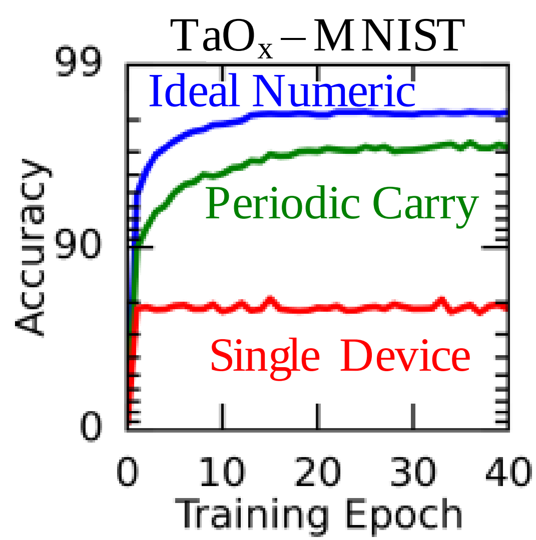
- Train with 1% signal
- Ideal result is 0.6

TaO_x Results



Carry once every
1000 updates

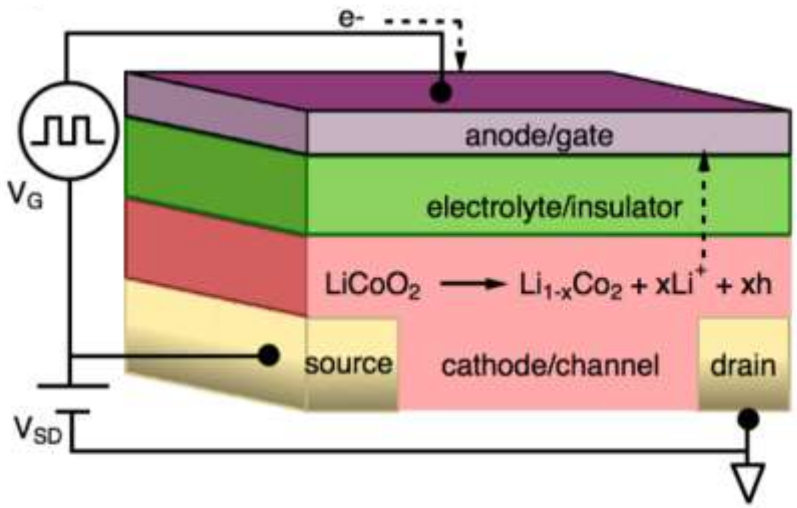


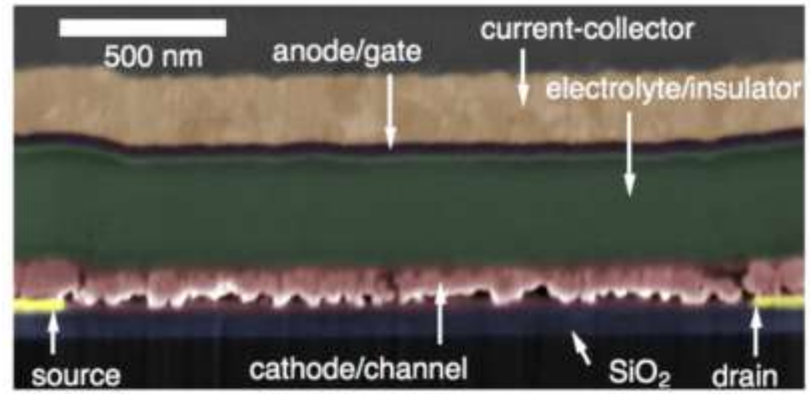


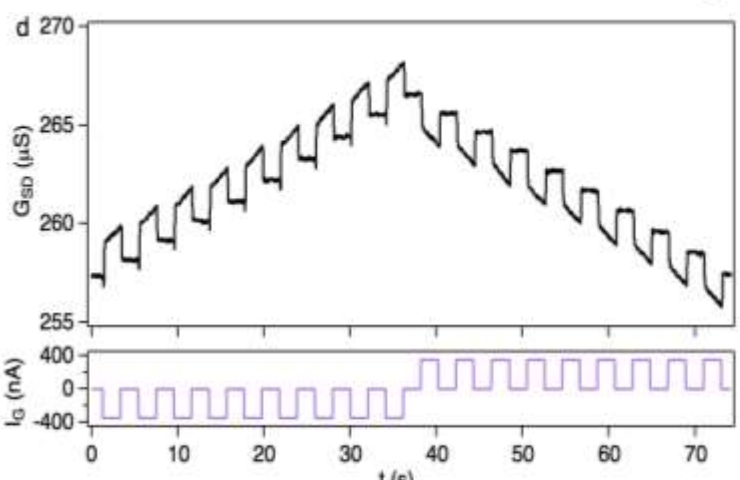
A/D and D/A is modeled, Serial operations modeled

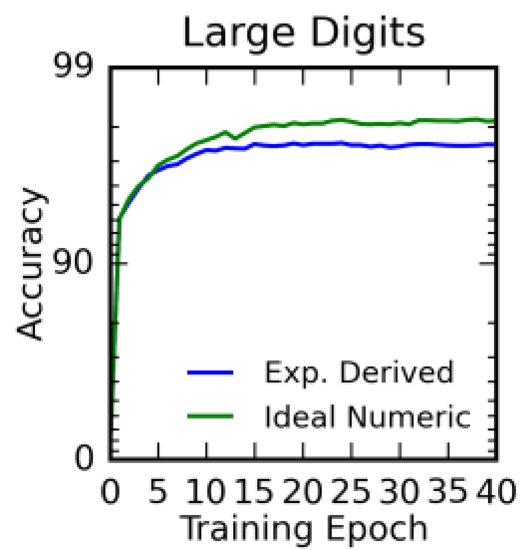
- When resetting weight, need to adjust pulse size based on current state to compensate for nonlinearity
- When reading a single weight, need to adjust readout range to be smaller (change capacitor on the integrator)

Li-Ion Synaptic Transistor for Analog Computation (LISTA)





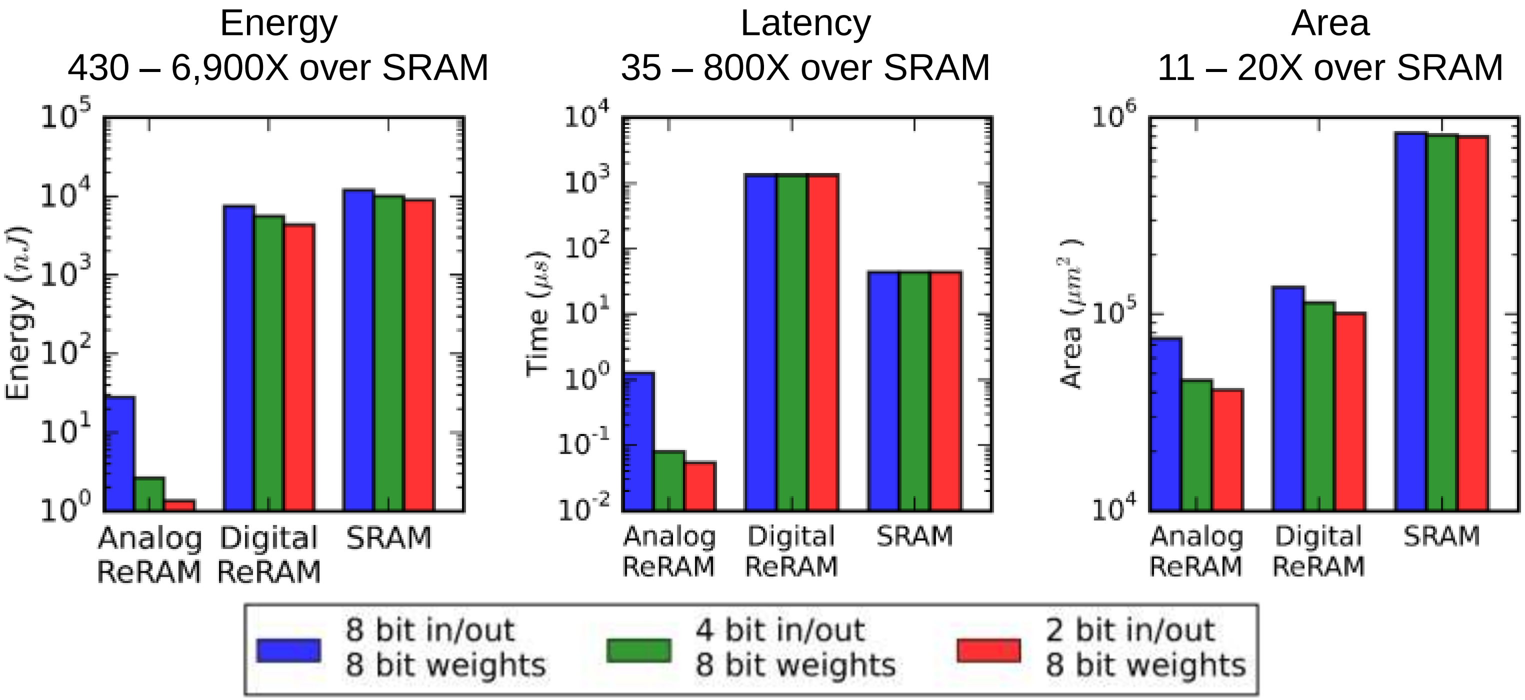




Off by ~ 1% from Ideal

E. J. Fuller, et al, "Li-Ion Synaptic Transistor for Low Power Analog Computing," *Advanced Materials,* vol. 29, no. 4, p. 1604310, 2017.

Summary



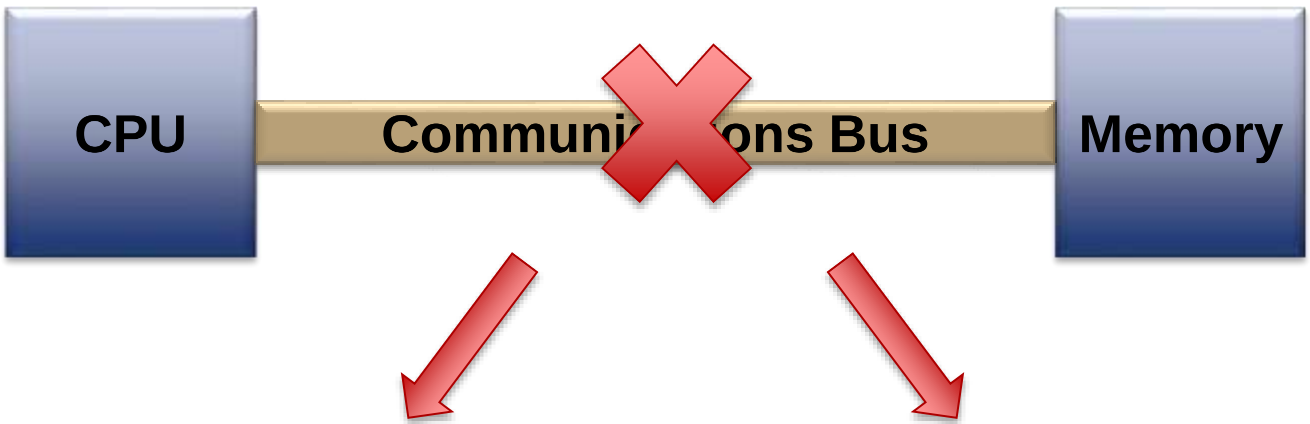
- Fundamental O(N) energy scaling advantage
- Use CrossSim to co-design materials to algorithms
  - Use periodic carry to overcome noise devices
- Need high resistance 10-100 MΩ Devices
- Need low write nonlinearities

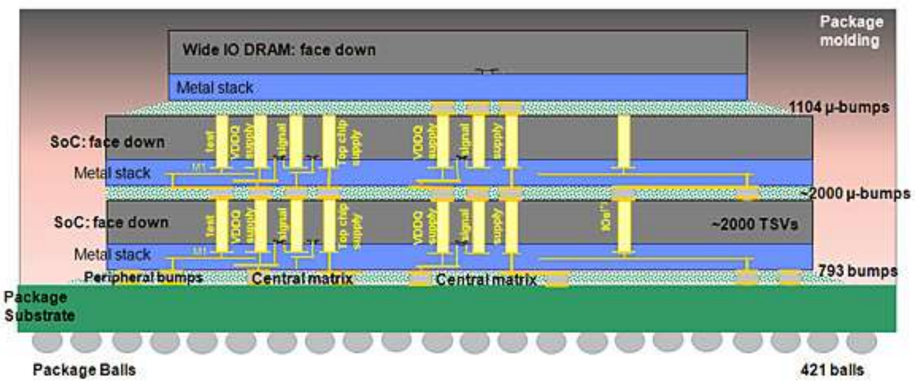
CROSS SIM

https://cross-sim.sandia.gov

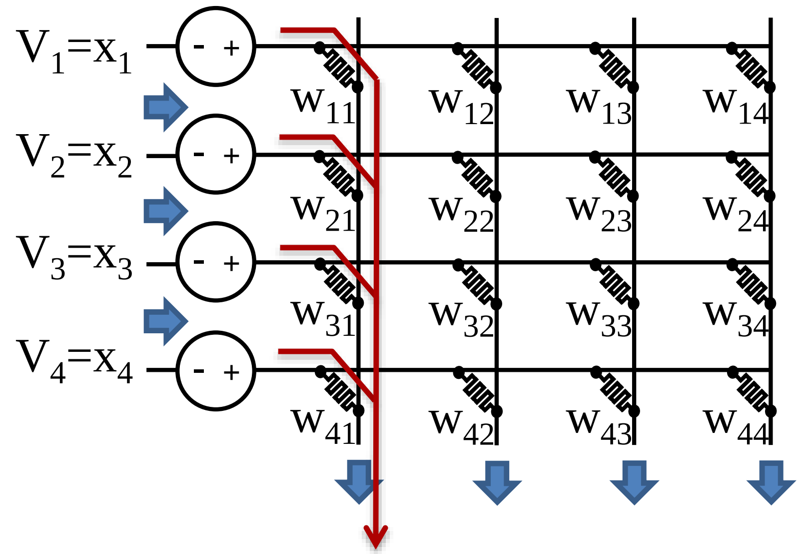
Extra Slides

Overcoming the Power Limit



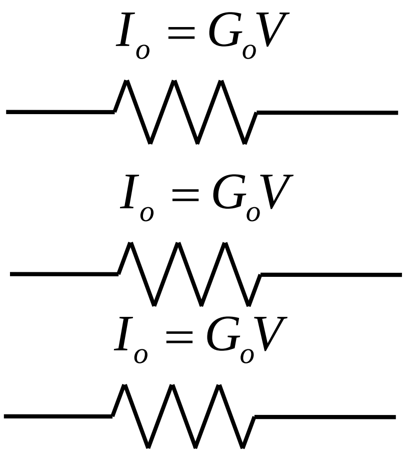


Richard Goering, "Three Die Stack -- A Big Step "Up" for 3D-ICs with TSVs" Cadence blog



Integrate Processing and Memory

The Noise Limited Energy to Read a Crossbar Column is Independent of Crossbar Size



Measure N resistors and determine the total output current with some signal to noise ratio (SNR)*

What is the minimum energy?

$$Energy = \underbrace{V^2 G_O \times N}_{\text{Power in each resistor} \times \text{number of resistors}} \times \frac{1}{\Delta f}$$

Determined by noise and SNR

$$\text{Thermal Noise} = \langle \Delta I^2 \rangle$$
$$= N \times (4k_b T \times G_o \times \Delta f)$$

$$SNR^2 = \frac{(NI_o)^2}{\langle \Delta I^2 \rangle}$$

$$\frac{1}{\Delta f} = 4k_b T \times SNR^2 \times \frac{1}{V^2 G_o \times N}$$

If we double the number of resistors, we can double the speed to get the same energy and SNR.

This is because the noise scales as sqrt(N) while the signal scales as N

$$Energy = 4k_b T \times SNR^2$$

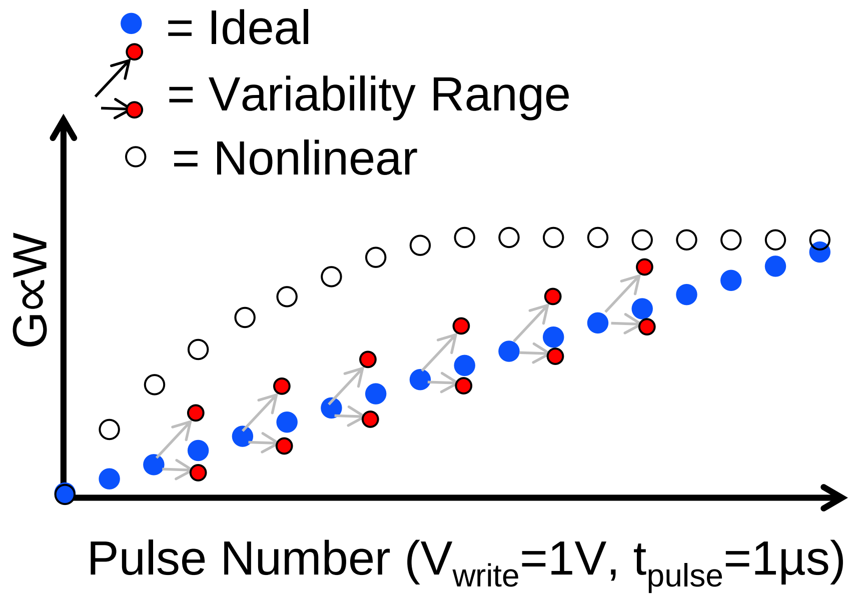
*we are assuming we need some fixed precision on the output, and don't need full floating point accuracy

Experimental Device Non-idealities

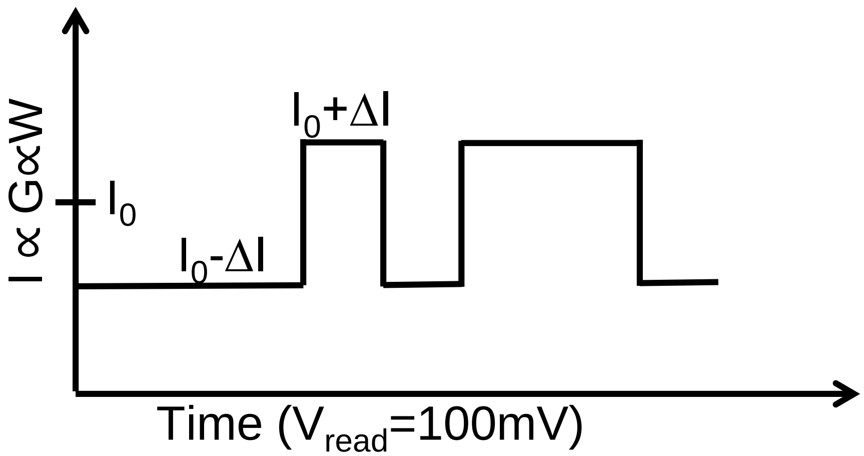
Device: **Write Variability, Write Nonlinearity, Asymmetry, Read Noise**

Circuit: A/D, D/A noise, parasitics

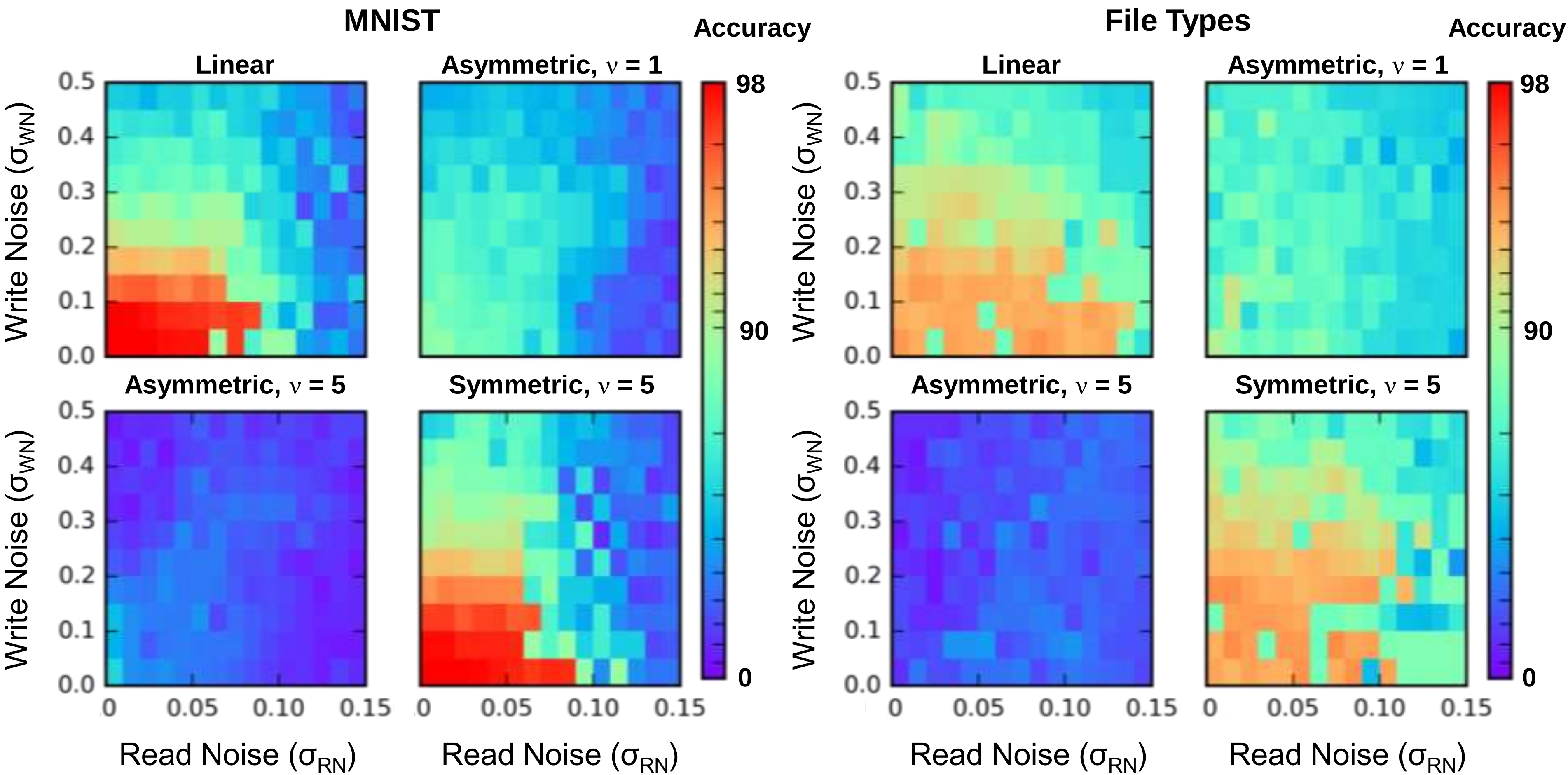
Variability and Nonlinearity



Read Noise

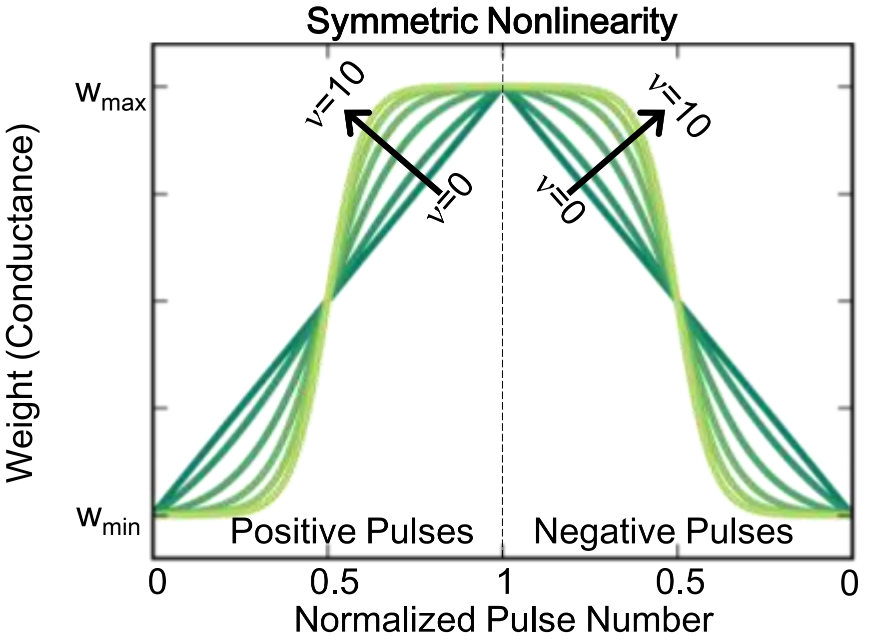


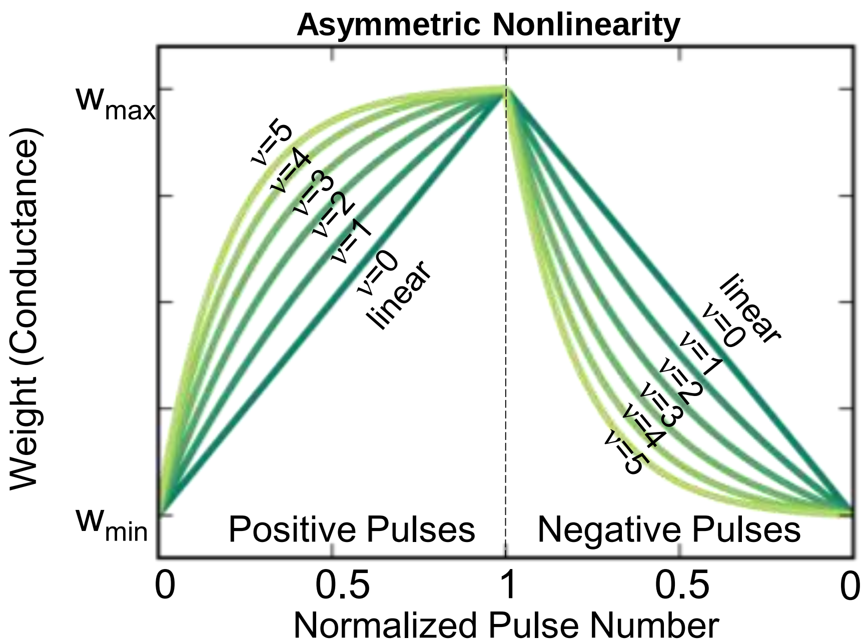
Combined Effects of Nonidealities



What are the Neural ReRAM Device Requirements?

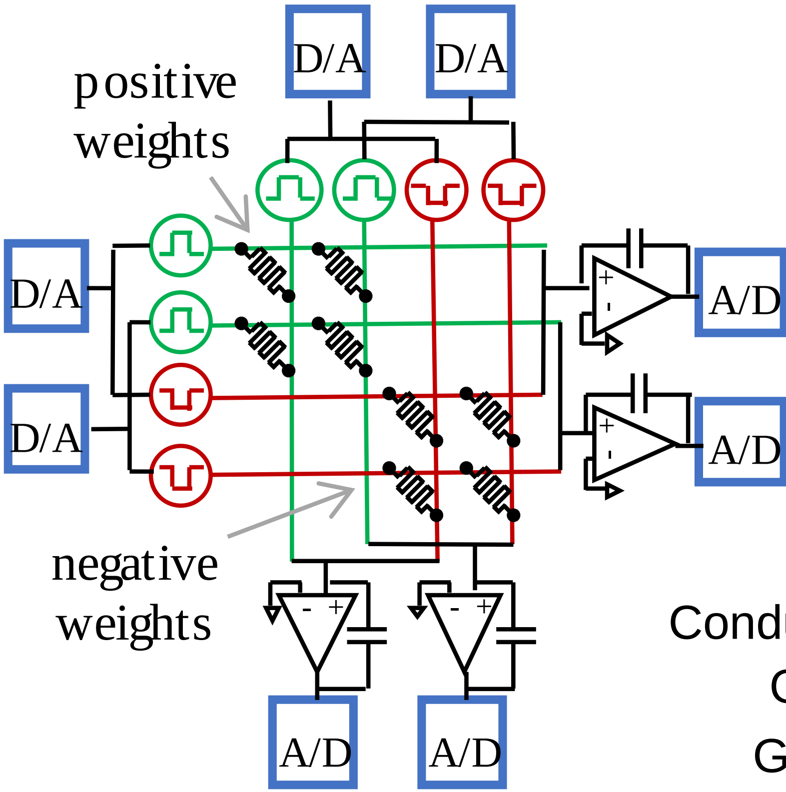
	Small Images	Large Images	File Types
Read Noise σ (% Range)	3%	5%	9%
Write Noise σ (% Range)	0.3%	0.4%	0.4%
Asymmetric Nonlinearity (v)	0.1	0.1	0.1
Symmetric Nonlinearity (v)	>20	5	5
Maximum Current	160 nA	13 nA	40 nA

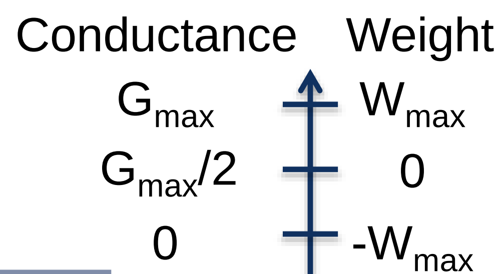


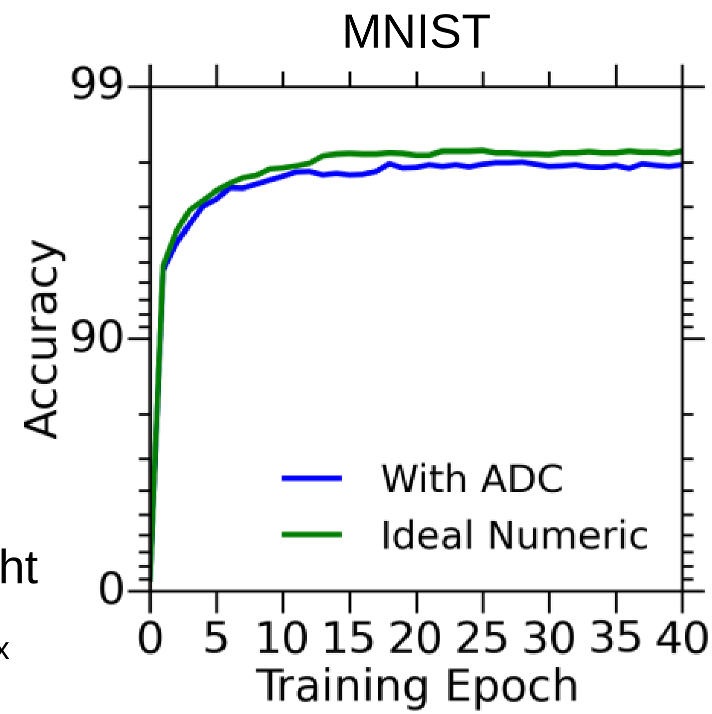


Full System Simulation

A/D & D/A Have Minimal Impact



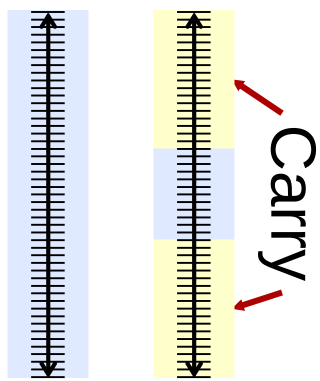


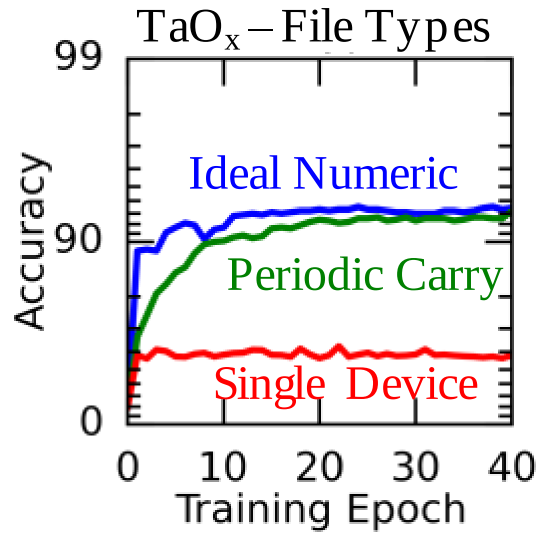


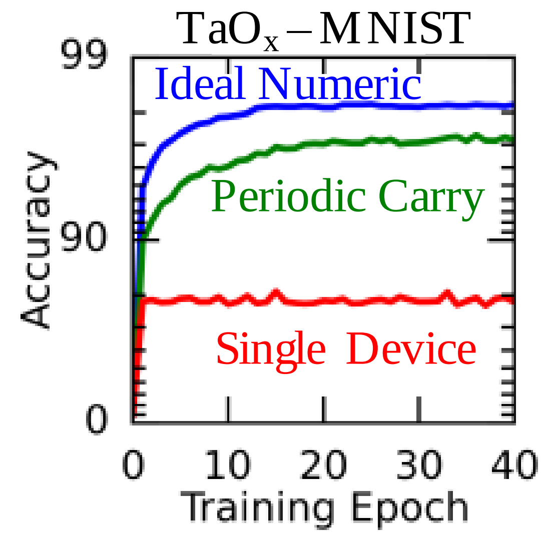
	Range	Bits
Row Input	-1 to 1	8
Col Output	-6 to 6	8
Col Input	-1 to 1	8
Row Output	-4 to 4	8
Row Update	-0.01 to 0.01	7
Col Update	-1 to 1	5

Data set	#Training/Test Examples	Network Size
File Types	4,501 / 900	256×512×9
MNIST	60,000 /10,000	784×300×10

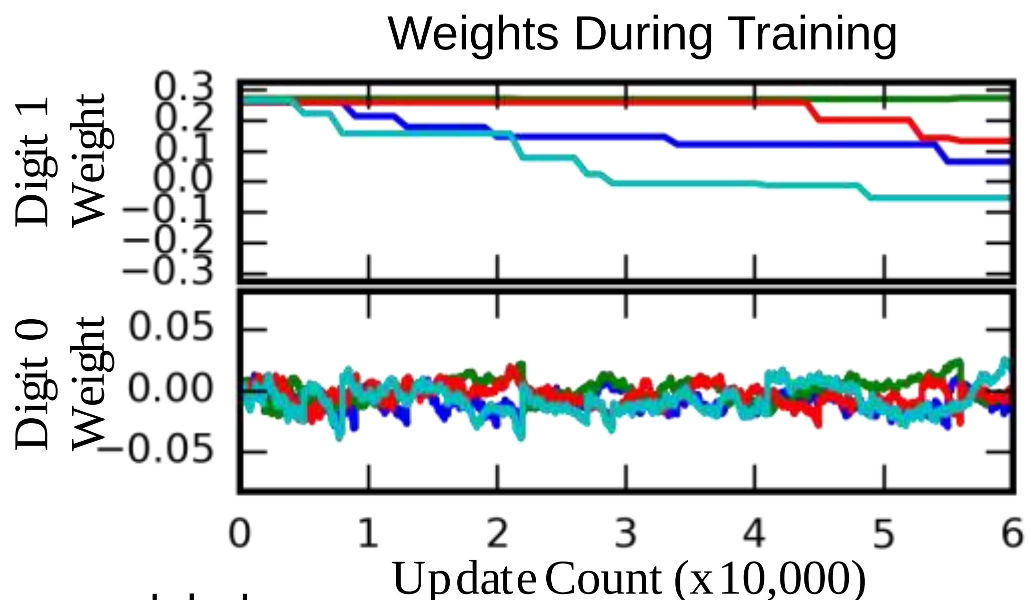
TaO$_x$ Results







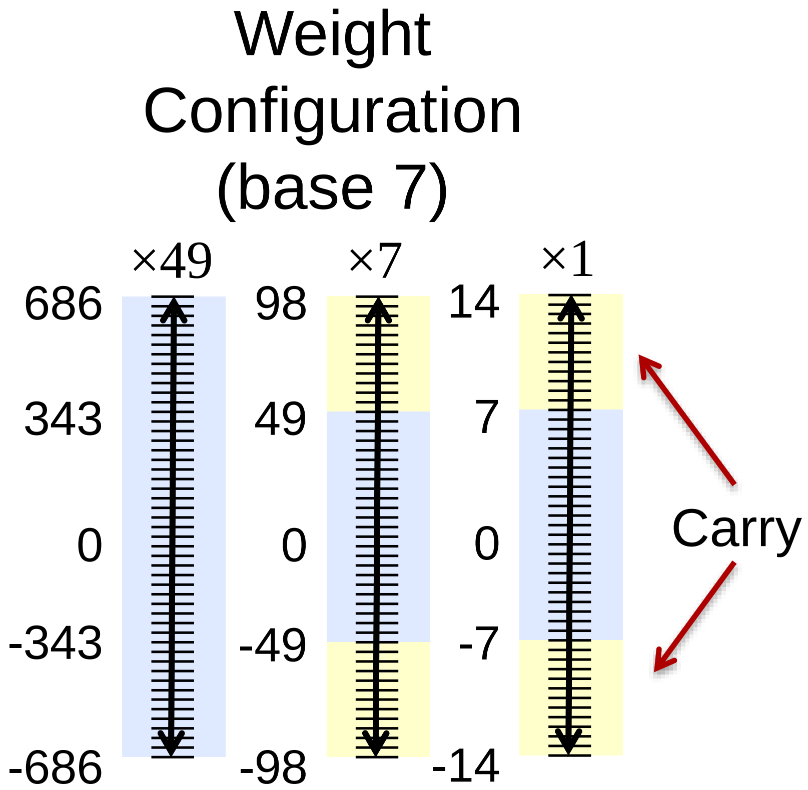
Carry once every 1000 updates for the LSB, and every 2 updates on others

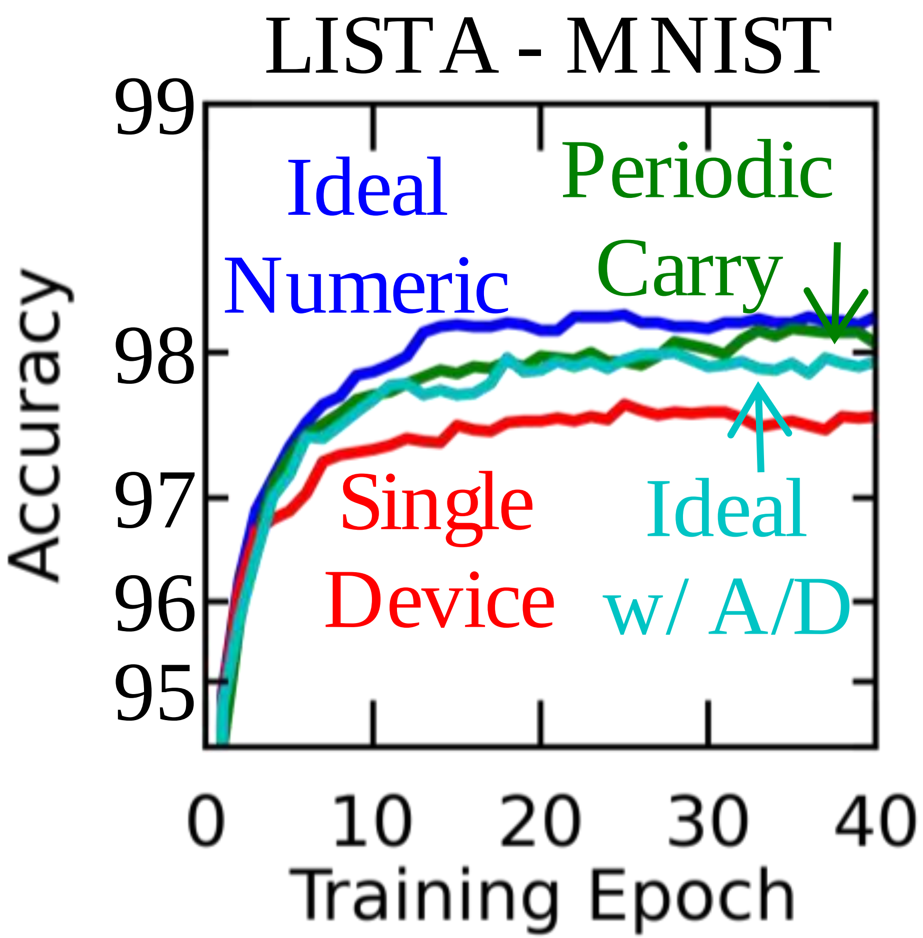


A/D and D/A is modeled, serial operations modeled

- When resetting weight, need to adjust pulse size based on current state to compensate for nonlinearity
- When reading a single weight, need to adjust readout range to be smaller (change capacitor on the integrator)

LISTA Results





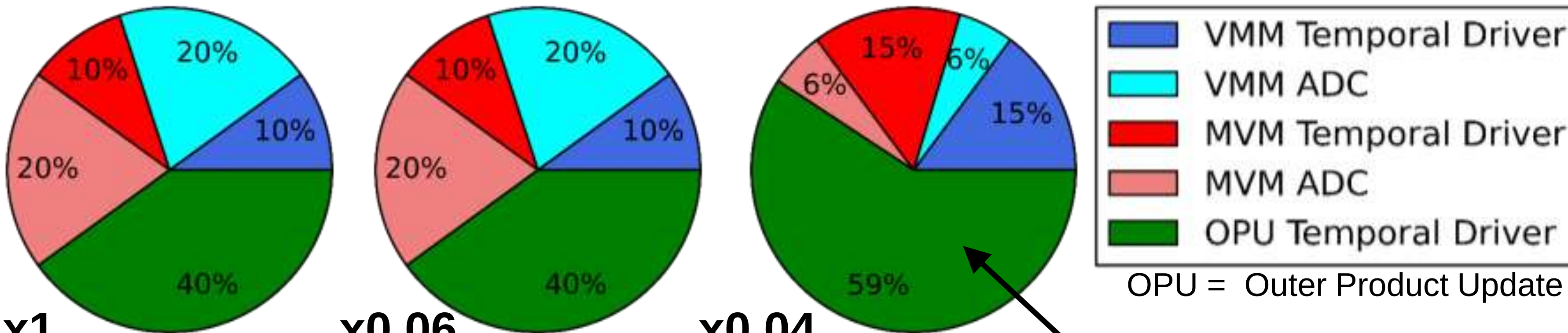
- Carry once every 1000 updates
- Use a single device per weight and subtract a reference current

Neural Core Latency Analysis

Sandia National Laboratories

Analog ReRAM

8 bit in/out | 4 bit in/out | 2 bit in/out



OPU = Outer Product Update

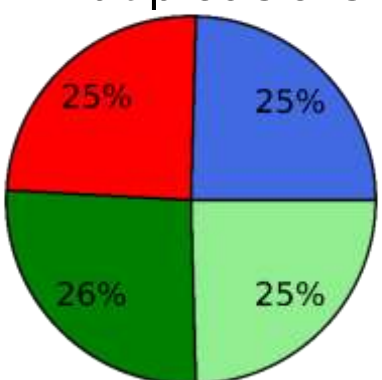
x1 1.28 µs

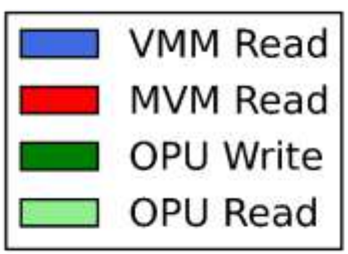
x0.06 0.08 µs

x0.04 0.054 µs

Min write time of 8 ns vs 1 ns incremental write

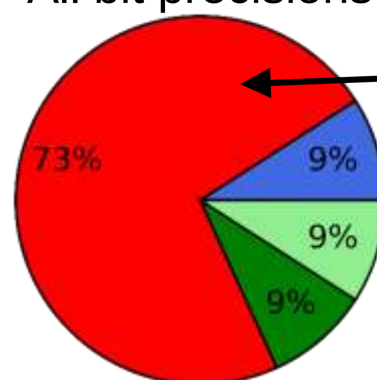
Digital ReRAM
All bit precisions





x1040 1335 µs

SRAM
All bit precisions



SRAM transpose read expensive

x35 44 µs

Neural Core Area Analysis

	8 bits In/out 8 bit weights	4 bits In/out 8 bit weights	2 bits In/out 8 bit weights

Analog ReRAM

x1

75k μm^2

Digital ReRAM

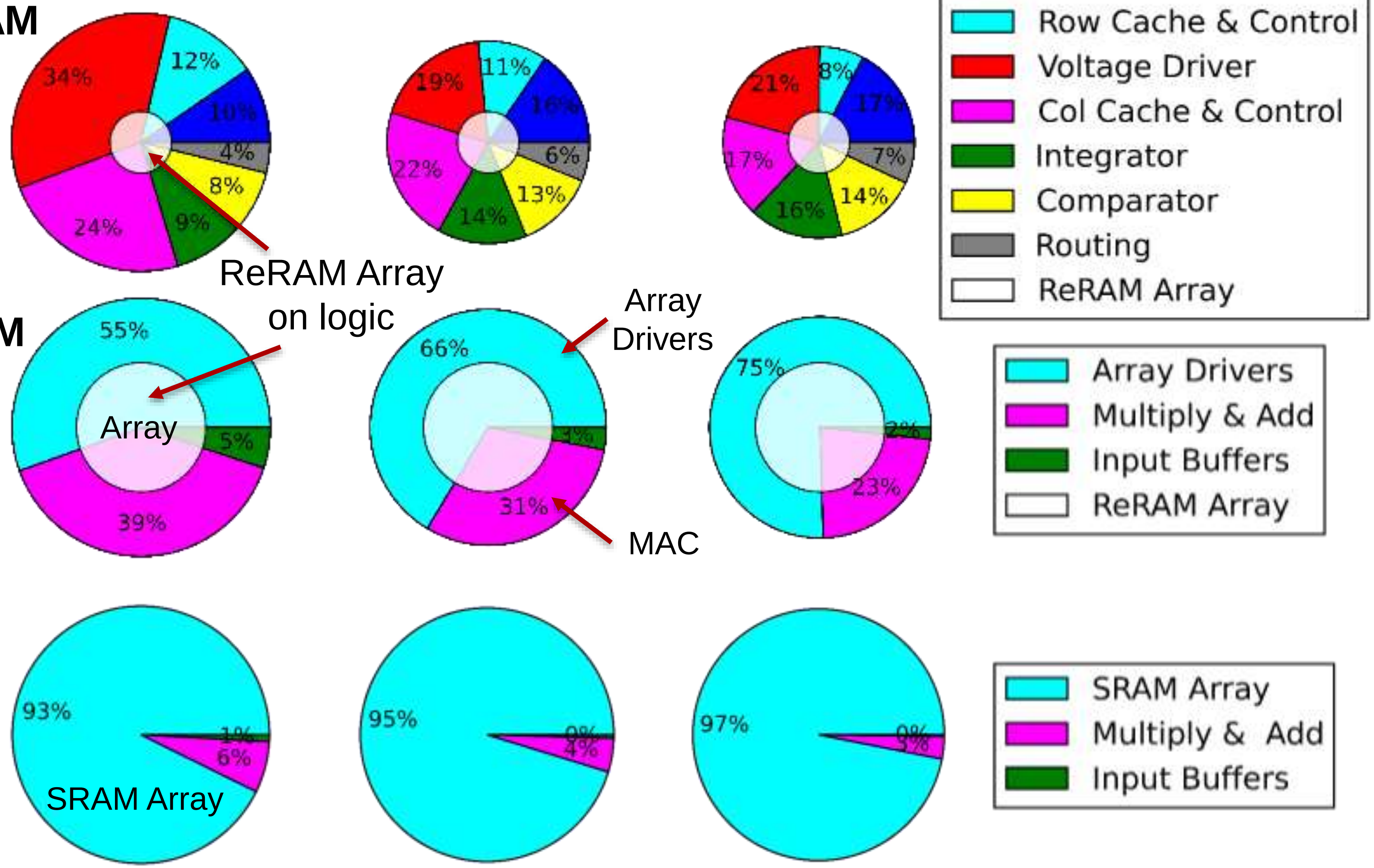
x1.8

137k μm^2

SRAM

x11.1

836k μm^2



For the ReRAM, high voltage transistors require 8X area, improving this could give ~2X area savings